EXPONENTIAL ROSENBROCK-TYPE METHODS

MARLIS HOCHBRUCK*, ALEXANDER OSTERMANN**, AND JULIA SCHWEITZER*

Abstract. We introduce a new class of exponential integrators for the numerical integration of large-scale systems of stiff differential equations. These so-called Rosenbrock-type methods linearize the flow in each time step and make use of the matrix exponential and related functions of the Jacobian. In contrast to standard integrators, the methods are fully explicit and do not require the numerical solution of linear systems. We analyze the convergence properties of these integrators in a semigroup framework of semilinear evolution equations in Banach spaces. In particular, we derive an abstract stability and convergence result for variable step sizes. This analysis further provides the required order conditions and thus allows us to construct pairs of embedded methods. We present a third order method with two stages, and a fourth order method with three stages, respectively. The application of the required matrix functions to vectors are computed by Krylov subspace approximations. We briefly discuss these implementation issues, and we give numerical examples that demonstrate the efficiency of the new integrators.

Key words. Exponential Rosenbrock-type methods, exponential integrators, stiff order conditions, stability bounds, convergence bounds, embedded methods of high order, variable step size implementation

AMS subject classifications. 65M12, 65L06

1. Introduction.

In this paper, we are concerned with a new class of numerical methods for the time integration of large systems of stiff differential equations

$$u'(t) = F(t, u(t)), \qquad u(t_0) = u_0. \tag{1.1}$$

Such equations typically arise from spatial discretizations of nonlinear time dependent partial differential equations. The numerical work when solving (1.1) by standard integrators like implicit Runge–Kutta methods or backward differentiation formulas (BDF) is often dominated by the numerical linear algebra which is required for the solution of the arising nonlinear systems of equations. For a collection of ode solvers, test problems and related references we refer to [21]. In particular, we point out the codes `VODEPK` [1, 2] and `ROWMAP` [28] where the linear algebra is based on Krylov subspace methods. Runge–Kutta discretizations of nonlinear evolution equations have been studied in [19, 20, 22].

Exponential integrators, on the other hand, require the matrix exponential and related functions of a certain matrix. Most exponential integrators analyzed so far in literature [5, 6, 9, 14, 16, 17, 18, 23, 26] make use of a (rough) a priori linearization

$$u'(t) = Au(t) + f(t, u(t)) \tag{1.2}$$

of the nonlinear problem (1.1). The matrix A then *explicitly* enters the formulation of the exponential integrator as the argument where the matrix functions are evaluated. Such an approach is justified in situations where the remainder f is small, or at least bounded in terms of A. The latter is the case for semilinear parabolic problems, if f is relatively bounded with respect to A. In particular, if A has a simple structure,

*Mathematisches Institut, Heinrich-Heine Universität Düsseldorf, Universitätsstr. 1, D–40225 Düsseldorf, Germany (E-mail: {marlis,schweitzer}@am.uni-duesseldorf.de)

**Institut für Mathematik, Universität Innsbruck, Technikerstr. 13, A–6020 Innsbruck, Austria (E-mail: alexander.ostermann@uibk.ac.at)

This work has been supported by the Deutsche Forschungsgemeinschaft through the Transregio-SFB 18.

it is possible to compute the product of a matrix function with a vector in a fast and reliable way. For instance, if A is the semi-discretization of the Laplacian on a regular rectangular mesh, these functions can be computed by fast Fourier transform techniques. Such an approach has been used in [16].

On the other hand, a fixed linearization like (1.2) can also lead to problems. As the remainder f is integrated explicitly by standard exponential methods, a badly chosen linearization can cause a severe step size restriction. This, for instance, is the case if the numerical solution stays near an equilibrium point (e.g., a saddle point) of the problem for a long time. If the linearization (1.2) is performed far from this equilibrium point, the integrator is forced to take small steps due to stability requirements. This will cause computational inefficiency.

In order to avoid these problems, we propose a new class of exponential integrators that linearize (1.1) in each integration step. The linearization can be computed either analytically or numerically. We first presented this approach in [15]. Here we give a rigorous stability and convergence proof, we discuss a possible variable step size implementation, and we give numerical comparisons. Related ideas have been used in [12] and [27]. Since the Jacobian of the problem changes from step to step, FFT techniques can no longer be used to compute the products of matrix functions with vectors. We will use Krylov subspace approximations instead [7, 11].

The outline of our paper is as follows. In Section 2, we introduce the method class and discuss a reformulation of the method which allows an efficient implementation with Krylov subspace methods. An implementation using Leja points was proposed in [3]. Since the reformulation speeds up the Krylov implementation considerably, we will not consider Leja point methods in this paper. In Section 3, we introduce the analytic framework and derive preliminary error bounds. We work in a framework of C_0 semigroups that covers many abstract semilinear evolution equations in Banach spaces. In contrast to exponential Runge–Kutta methods [14], the new class of Rosenbrock-type methods produces smaller defects when inserting the exact solution into the numerical scheme. This is due to the linearization. It facilitates the derivation of the order conditions and gives much simpler conditions than in [14]. In particular, it is possible to construct a fourth order integrator with an embedded third order method, using three stages only. Since the Jacobian varies from step to step, the stability estimate of the discrete evolution operator is crucial. The necessary stability bounds for variable step size discretizations are derived in Section 3.3.

In Section 4, we give a convergence bound for methods up to order 4. Particular methods of order three and four are given in Section 5, a generalization to non-autonomous problems is discussed in Section 6. In Section 7, we briefly describe an implementation based on Krylov subspace approximations, and we present two numerical examples: a two dimensional advection-diffusion-reaction problem and a Schrödinger equation with time dependent potential. The possible extensions for analytic semigroups is sketched in the Appendix.

2. Exponential Rosenbrock-type methods.

In this paper we consider the time discretization of (possibly abstract) differential equations in autonomous form

$$u'(t) = F(u(t)), \qquad u(t_0) = u_0. \tag{2.1}$$

The precise assumptions on the problem class will be stated in Section 3 below. The numerical schemes considered are based on a continuous linearization of (2.1) along the numerical solution. For a given point u_n in the state space, this linearization is

$$u'(t) = J_n u(t) + g_n(u(t)), \tag{2.2a}$$

$$J_n = \mathrm{D}F(u_n) = \frac{\partial F}{\partial u}(u_n), \qquad g_n(u(t)) = F(u(t)) - J_n u(t) \tag{2.2b}$$

with J_n denoting the Jacobian of F, and g_n the nonlinear remainder, evaluated at u_n, respectively. The numerical schemes will make *explicit* use of these quantities.

2.1. Method class. Let u_n denote the numerical approximation to the solution of (2.1) at time t_n. Its value at t_0 is given by the initial condition. Applying an explicit exponential Runge–Kutta scheme [14] to (2.2a), we obtain the following class of explicit one-step methods

$$U_{ni} = \mathrm{e}^{c_i h_n J_n} u_n + h_n \sum_{j=1}^{i-1} a_{ij}(h_n J_n) g_n(U_{nj}), \qquad 1 \le i \le s \tag{2.3a}$$

$$u_{n+1} = \mathrm{e}^{h_n J_n} u_n + h_n \sum_{i=1}^{s} b_i(h_n J_n) g_n(U_{ni}). \tag{2.3b}$$

Here, $h_n > 0$ denotes a positive time step, and u_{n+1} is the numerical approximation to the exact solution at time $t_{n+1} = t_n + h_n$.

The method is built on s internal stages U_{ni} that approximate the solution at $t_n + c_i h_n$. The real numbers c_i are called nodes of the method. The method is fully explicit and does not require the solution of linear or nonlinear systems of equations. As usual in exponential integrators, the weights $b_i(z)$ are linear combinations of the entire functions

$$\varphi_k(z) = \int_0^1 \mathrm{e}^{(1-\sigma)z} \frac{\sigma^{k-1}}{(k-1)!} \, \mathrm{d}\sigma, \quad k \ge 1. \tag{2.4}$$

These functions satisfy the recurrence relations

$$\varphi_k(z) = \frac{\varphi_{k-1}(z) - \varphi_{k-1}(0)}{z}, \qquad \varphi_0(z) = \mathrm{e}^z. \tag{2.5}$$

The coefficients $a_{ij}(z)$ will be chosen as linear combinations of the related functions $\varphi_k(c_i z)$. Henceforth, the methods (2.3) will be called *exponential Rosenbrock methods.*

Without further mentioning, we will assume throughout the paper that the methods fulfill the following simplifying assumptions

$$\sum_{i=1}^{s} b_i(z) = \varphi_1(z), \qquad \sum_{j=1}^{i-1} a_{ij}(z) = c_i \varphi_1(c_i z), \quad 1 \le i \le s. \tag{2.6}$$

Note that (2.6) implies $c_1 = 0$ and consequently $U_{n1} = u_n$.

Methods that satisfy the simplifying assumptions (2.6) possess several interesting features. They preserve equilibria of (2.1), they have small defects which in turn lead to simple order conditions for stiff problems (Section 3.1), they allow a reformulation for efficient implementation, see below, and they can easily be extended to non-autonomous problems (Section 6).

2.2. Reformulation of the method. For the implementation of an exponential Rosenbrock method, it is crucial to approximate the application of matrix functions to vectors efficiently. We therefore suggest to express the vectors $g_n(U_{nj})$ as

$$g_n(U_{nj}) = g_n(u_n) + D_{nj}, \qquad 2 \le j \le s.$$

A similar approach was used in [27]. Due to the simplifying assumptions (2.6), the method (2.3) takes the equivalent form

$$U_{ni} = u_n + c_i h_n \varphi_1(c_i h_n J_n) F(u_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}, \tag{2.7a}$$

$$u_{n+1} = u_n + h_n \varphi_1(h_n J_n) F(u_n) + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni}. \tag{2.7b}$$

The main motivation for this reformulation is that the vectors D_{ni} are expected to be small in norm. When computing the application of matrix functions to these vectors with some Krylov subspace method, this should be possible in a low dimensional subspace. Consequently, only one computationally expensive Krylov approximation will be required in each time step, namely that involving $F(u_n)$. A similar idea has also been used to make the code `exp4` efficient [12].

3. Analytic framework and preliminary error analysis. For the error analysis of (2.3), we work in a semigroup framework. Background information on semigroups can be found in the textbooks [8, 24]. Let

$$J = J(u) = \mathrm{D}F(u) = \frac{\partial F}{\partial u}(u) \tag{3.1}$$

be the Fréchet derivative of F in a neighborhood of the exact solution of (2.1). Throughout the paper we consider the following assumptions.

ASSUMPTION C.1. *The linear operator J is the generator of a strongly continuous semigroup e^{tJ} on a Banach space X. More precisely, we assume that there exist constants C and ω such that*

$$\left\| \mathrm{e}^{tJ} \right\|_{X \leftarrow X} \le C\, \mathrm{e}^{\omega t}, \qquad t \ge 0 \tag{3.2}$$

holds uniformly in a neighborhood of the exact solution of (2.1).

Recall that the analytic functions $b_i(z)$ and $a_{ij}(z)$ are linear combinations of $\varphi_k(z)$ and $\varphi_k(c_i z)$, respectively. These functions are related to the exponential function through (2.4). Assumption C.1 thus guarantees that the coefficients $b_i(hJ)$ and $a_{ij}(hJ)$ of the method are bounded operators. This property is crucial in our proofs.

In the subsequent analysis we restrict our attention to semilinear problems

$$u'(t) = F(u(t)), \qquad F(u) = Au + f(u), \qquad u(t_0) = u_0. \tag{3.3}$$

This implies that (2.2b) takes the form

$$J_n = A + \frac{\partial f}{\partial u}(u_n), \qquad g_n(u(t)) = f(u(t)) - \frac{\partial f}{\partial u}(u_n) u(t). \tag{3.4}$$

Our main hypothesis on the nonlinearity f is the following:

Assumption C.2. *We suppose that* (3.3) *possesses a sufficiently smooth solution* $u : [0, T] \to X$ *with derivatives in* X, *and that* $f : X \to X$ *is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are supposed to be uniformly bounded.*

By Assumption C.2, the Jacobian (3.1) satisfies the Lipschitz condition

$$\|J(u) - J(v)\|_{X \leftarrow X} \le C\,\|u - v\| \tag{3.5}$$

in a neighborhood of the exact solution.

Remark. If the semigroup generated by J is not only strongly continuous but analytic, more general nonlinearities can be analyzed. To keep our presentation simple, we restrict ourselves to strongly continuous semigroups for the moment and sketch the possible extensions to analytic semigroups later in Appendix A.

Examples will be considered in Section 7.

3.1. Defects. For brevity, we denote $G_n(t) = g_n(u(t))$. Inserting the exact solution into the numerical scheme gives

$$u(t_n + c_i h_n) = \mathrm{e}^{c_i h_n J_n} u(t_n) + h_n \sum_{j=1}^{i-1} a_{ij}(h_n J_n) G_n(t_n + c_j h_n) + \Delta_{ni}, \tag{3.6a}$$

$$u(t_{n+1}) = \mathrm{e}^{h_n J_n} u(t_n) + h_n \sum_{i=1}^{s} b_i(h_n J_n) G_n(t_n + c_i h_n) + \delta_{n+1} \tag{3.6b}$$

with defects Δ_{ni} and δ_{n+1}. The computation and estimation of the defects is carried out in the same way as in our previous paper [14, Sect. 4.1]. In particular, expressing the left-hand side of (3.6a) by the variation-of-constants formula

$$u(t_n + c_i h_n) = \mathrm{e}^{c_i h_n J_n} u(t_n) + \int_0^{c_i h_n} \mathrm{e}^{(c_i h_n - \tau) J_n} G_n(t_n + \tau)\,\mathrm{d}\tau$$

and then expanding G_n into a Taylor series at t_n yields

$$\Delta_{ni} = h_n \psi_{1,i}(h_n J_n) G_n(t_n) + h_n^2 \psi_{2,i}(h_n J_n) G_n'(t_n) + \Delta_{ni}^{[2]}, \tag{3.7}$$

with

$$\psi_{j,i}(z) = \varphi_j(c_i z) c_i^j - \sum_{k=1}^{i-1} a_{ik}(z) \frac{c_k^{j-1}}{(j-1)!} \tag{3.8}$$

and remainders $\Delta_{ni}^{[2]}$ satisfying

$$\big\|\Delta_{ni}^{[2]}\big\| \le C h_n^3. \tag{3.9}$$

Small defects in the internal stages facilitate our convergence proofs considerably. This gives a further reason for requiring (2.6) which implies $\psi_{1,i}(z) \equiv 0$. Unfortunately, explicit methods *cannot* have $\psi_{2,i}(z) \equiv 0$ for all i. Nevertheless, the second term on the right-hand side of (3.7) turns out to be small. This is seen from the identity

$$G_n'(t_n) = \frac{\partial g_n}{\partial u}\big(u(t_n)\big) u'(t_n) = \left(\frac{\partial f}{\partial u}\big(u(t_n)\big) - \frac{\partial f}{\partial u}(u_n)\right) u'(t_n),$$

No.	condition in defect	order condition	order
1	$\psi_1(z) \equiv 0$	$\sum_{i=1}^{s} b_i(z) = \varphi_1(z)$	1
2	$\psi_{1,i}(z) \equiv 0$	$\sum_{j=1}^{i-1} a_{ij}(z) = c_i \varphi_1(c_i z), \quad 2 \le i \le s$	2
3	$\psi_3(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z) c_i^2 = 2\varphi_3(z)$	3
4	$\psi_4(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z) c_i^3 = 6\varphi_4(z)$	4

TABLE 3.1
Stiff order conditions for exponential Rosenbrock methods applied to autonomous problems.

which itself is a consequence of linearizing at each step, cf. (3.4). By Assumption C.2 this relation implies

$$\|G_n'(t_n)\| \le C\|e_n\| \tag{3.10}$$

with $e_n = u_n - u(t_n)$, and the defects of the internal stages thus obey the bound

$$\|\Delta_{ni}\| \le Ch_n^2\|e_n\| + Ch_n^3. \tag{3.11}$$

Similarly, we get for the defects δ_{n+1} at time t_{n+1}

$$\delta_{n+1} = \sum_{j=1}^{q} h_n^j \psi_j(h_n J_n) G_n^{(j-1)}(t_n) + \delta_{n+1}^{[q]}, \tag{3.12}$$

with

$$\psi_j(z) = \varphi_j(z) - \sum_{k=1}^{s} b_k(z) \frac{c_k^{j-1}}{(j-1)!} \tag{3.13}$$

and remainders $\delta_{n+1}^{[q]}$ satisfying

$$\|\delta_{n+1}^{[q]}\| \le Ch_n^{q+1}. \tag{3.14}$$

Again, small defects are desirable. Due to (2.6), we have $\psi_1(z) \equiv 0$. To obtain higher order bounds for δ_{n+1} first observe that the h^2-term in (3.12) is small due to (3.10). Additional terms vanish if $\psi_j = 0$, $j \ge 3$.

All conditions encountered so far are collected in Table 3.1. They will later turn out to be the order conditions for methods up to order 4.

LEMMA 3.1. *If the order conditions of Table* 3.1 *are satisfied up to order* $p \le 4$, *we obtain*

$$\|\delta_{n+1}\| \le Ch_n^2\|e_n\| + Ch_n^{p+1}. \tag{3.15}$$

Proof. This at once follows from (3.12). □

3.2. Preliminary error bounds. Let

$$e_n = u_n - u(t_n) \quad \text{and} \quad E_{ni} = U_{ni} - u(t_n + c_i h_n)$$

denote the differences between the numerical solution and the exact solution. Subtracting (3.6) from the numerical method (2.3) gives the error recursion

$$E_{ni} = \mathrm{e}^{c_i h_n J_n} e_n + h_n \sum_{j=1}^{i-1} a_{ij}(h_n J_n)\Big(g_n(U_{nj}) - G_n(t_n + c_j h_n)\Big) - \Delta_{ni}, \tag{3.16a}$$

$$e_{n+1} = \mathrm{e}^{h_n J_n} e_n + h_n \sum_{i=1}^{s} b_i(h_n J_n)\Big(g_n(U_{ni}) - G_n(t_n + c_i h_n)\Big) - \delta_{n+1}. \tag{3.16b}$$

We will derive bounds for these errors.

Lemma 3.2. *Under Assumption* C.2*, we have*

$$\|g_n(U_{ni}) - G_n(t_n + c_i h_n)\| \le C\big(h_n + \|e_n\| + \|E_{ni}\|\big) \|E_{ni}\|\,, \tag{3.17a}$$

$$\|g_n(u_n) - G_n(t_n)\| \le C \|e_n\|^2\,, \tag{3.17b}$$

$$\Big\| \frac{\partial g_n}{\partial u}\big(u(t_n)\big) \Big\|_{X \leftarrow X} \le C \|e_n\|\,, \tag{3.17c}$$

as long as the errors E_{ni} and e_n remain in a sufficiently small neighborhood of 0.

Proof. The last bound (3.17c) is a direct consequence of the linearization and the Lipschitz condition (3.5). Using Taylor series expansion, we get

$$\begin{aligned} g_n(U_{ni}) - G_n(t_n + c_i h_n) &= \frac{\partial g_n}{\partial u}\big(u(t_n + c_i h_n)\big) E_{ni} \\ &\quad + \int_0^1 (1-\tau) \frac{\partial^2 g_n}{\partial u^2}\big(u(t_n + c_i h_n) + \tau E_{ni}\big)(E_{ni}, E_{ni})\, \mathrm{d}\tau. \end{aligned}$$

Setting $i = 1$ at once proves (3.17b). To derive (3.17a), we expand the first term on the right-hand side once more at t_n and use the identity

$$\frac{\partial g_n}{\partial u}\big(u(t_n)\big) = -\int_0^1 \frac{\partial^2 g_n}{\partial u^2}\big(u(t_n) + \tau e_n\big) e_n \,\mathrm{d}\tau.$$

This finally proves (3.17a). □

Using this result, we can establish an error bound for the internal stages.

Lemma 3.3. *Under Assumptions* C.1 *and* C.2 *we have*

$$\|E_{ni}\| \le C \|e_n\| + C h_n^3,$$

as long as the global errors e_n remain in a bounded neighborhood of 0.

Proof. The assertion at once follows from (3.16a), Lemma 3.2, and (3.11). □

3.3. Stability bounds. In order to establish convergence bounds, we have to solve recursion (3.16b). For this purpose, stability bounds for the discrete evolution operators are crucial. In a first step we will show stability along the exact solution.

We commence with two auxiliary results.

Lemma 3.4. *Let the initial value problem* (3.3) *satisfy Assumptions* C.1 *and* C.2, *and let* $\widehat{J}_n = \mathrm{D}F(u(t_n))$. *Then, for any* $\widetilde{\omega} > \omega$, *there exists a constant* C_{L} *independent of* h_{n-1} *such that*

$$\left\| \mathrm{e}^{t\widehat{J}_n} - \mathrm{e}^{t\widehat{J}_{n-1}} \right\|_{X\leftarrow X} \le C_{\mathrm{L}} h_{n-1} \mathrm{e}^{\widetilde{\omega} t}, \qquad t \ge 0. \tag{3.18}$$

Proof. Applying the variation-of-constants formula to the initial value problem

$$v'(t) = \widehat{J}_n v(t) = \widehat{J}_{n-1} v(t) + \big(\widehat{J}_n - \widehat{J}_{n-1}\big) v(t)$$

shows the representation

$$\mathrm{e}^{t\widehat{J}_n} - \mathrm{e}^{t\widehat{J}_{n-1}} = \int_0^1 t \mathrm{e}^{(1-\sigma)t\widehat{J}_{n-1}} \big(\widehat{J}_n - \widehat{J}_{n-1}\big) \mathrm{e}^{\sigma t \widehat{J}_n} \mathrm{d}\sigma. \tag{3.19}$$

The required estimate now follows from (3.5) and the smoothness of $u(t)$. □

Lemma 3.5. *Under the assumptions of Lemma* 3.4, *the relation*

$$|||x|||_n = \sup_{t\ge 0} e^{-\widetilde{\omega} t} \left\| \mathrm{e}^{t\widehat{J}_n} x \right\|, \qquad x \in X \tag{3.20}$$

defines for any $n = 0, 1, 2, \ldots$ *a norm on* X. *This norm is equivalent to* $\|\cdot\|$ *and satisfies the bound*

$$|||x|||_n \le (1 + C_{\mathrm{L}} h_{n-1}) |||x|||_{n-1}, \qquad n \ge 1. \tag{3.21}$$

Proof. Obviously, we have $\|x\| \le |||x|||_n$. On the other hand, the bound (3.2) yields $|||x|||_n \le C\,\|x\|$. Thus, the two norms are equivalent.

For arbitrary $x \in X$, we have

$$\begin{aligned} |||x|||_n &= \sup_{t\ge 0} \mathrm{e}^{-\widetilde{\omega} t} \left\| \left(\mathrm{e}^{t\widehat{J}_n} - \mathrm{e}^{t\widehat{J}_{n-1}} + \mathrm{e}^{t\widehat{J}_{n-1}} \right) x \right\| \\ &\le |||x|||_{n-1} + \sup_{t\ge 0} \mathrm{e}^{-\widetilde{\omega} t} \left\| \mathrm{e}^{t\widehat{J}_n} - \mathrm{e}^{t\widehat{J}_{n-1}} \right\|_{X\leftarrow X} \|x\| \\ &\le (1 + C_{\mathrm{L}} h_{n-1})\, |||x|||_{n-1} \end{aligned}$$

by Lemma 3.4 and the equivalence of the norms. □

The following lemma proves the stability of the discrete evolution operators along the exact solution.

Lemma 3.6. *Under the assumptions of Lemma* 3.4, *there exists a constant* C *such that*

$$\left\| \mathrm{e}^{h_n \widehat{J}_n} \cdots \mathrm{e}^{h_0 \widehat{J}_0} \right\|_{X\leftarrow X} \;\le\; C\, \mathrm{e}^{\Omega(h_0 + \ldots + h_n)} \tag{3.22}$$

with $\Omega = C_{\mathrm{L}} + \widetilde{\omega}$.

Proof. By (3.20) and Lemma 3.5 we have

$$\begin{aligned} \left|\left|\left| \mathrm{e}^{h_n \widehat{J}_n} \cdots \mathrm{e}^{h_0 \widehat{J}_0} x \right|\right|\right|_n &= \sup_{t\ge 0} \left\| \mathrm{e}^{-\widetilde{\omega} t} \mathrm{e}^{t\widehat{J}_n} \mathrm{e}^{-\widetilde{\omega} h_n} \mathrm{e}^{\widetilde{\omega} h_n} \mathrm{e}^{h_n \widehat{J}_n} \cdots \mathrm{e}^{h_0 \widehat{J}_0} x \right\| \\ &\le \sup_{t\ge 0} \left\| \mathrm{e}^{-\widetilde{\omega} t} \mathrm{e}^{t\widehat{J}_n} \mathrm{e}^{\widetilde{\omega} h_n} \mathrm{e}^{h_{n-1} \widehat{J}_{n-1}} \cdots \mathrm{e}^{h_0 \widehat{J}_0} x \right\| \\ &= \mathrm{e}^{\widetilde{\omega} h_n} \left|\left|\left| \mathrm{e}^{h_{n-1} \widehat{J}_{n-1}} \cdots \mathrm{e}^{h_0 \widehat{J}_0} x \right|\right|\right|_n \\ &\le \mathrm{e}^{\widetilde{\omega} h_n} (1 + C_{\mathrm{L}} h_{n-1}) \left|\left|\left| \mathrm{e}^{h_{n-1} \widehat{J}_{n-1}} \cdots \mathrm{e}^{h_0 \widehat{J}_0} x \right|\right|\right|_{n-1}. \end{aligned}$$

Thus, the estimate $1+C_{\mathrm{L}}h_{n-1} \le \mathrm{e}^{C_{\mathrm{L}}h_{n-1}}$ together with an induction argument proves the lemma. □

We now turn our attention to the operators $J_n = \mathrm{D}F(u_n)$ that result from the linearization process (2.2). These operators constitute an essential component of the numerical scheme (2.3). The triangle inequality shows that

$$\|u_n - u_{n-1}\| \le Ch_{n-1} + \|e_n\| + \|e_{n-1}\|. \tag{3.23}$$

We now repeat the above estimations with J_n in the role of $\widehat{J}_n$ and, in particular, use (3.23) in the proof of Lemma 3.4. This gives the following stability result for the discrete evolution operators on X.

Theorem 3.7. *Let the initial value problem* (3.3) *satisfy Assumptions* C.1 *and* C.2. *Then, for any $\widetilde{\omega} > \omega$, there exist constants C and $C_{\mathtt{E}}$ such that*

$$\left\|\mathrm{e}^{h_nJ_n}\cdots\mathrm{e}^{h_0J_0}\right\|_{X\leftarrow X} \;\le\; C\,\mathrm{e}^{\Omega(h_0+\ldots+h_n)+C_{\mathtt{E}}\sum_{j=1}^n\|e_j\|} \tag{3.24}$$

with $\Omega = C_{\mathrm{L}} + \widetilde{\omega}$. The bound holds as long as the numerical solution u_n stays in a sufficiently small neighborhood of the exact solution of (3.3). □

The stability bound (3.24) requires some attention. Strictly speaking, stability is only guaranteed if the term $\sum_{j=1}^n \|e_j\|$ is uniformly bounded in n for $t_0 \le t_n \le T$. This condition can be considered as a (weak) restriction on the employed step size sequence, see the discussion in Section 4 below.

4. Error bounds. We are now ready to present the main result of our paper. We will show that the conditions of Table 3.1 are sufficient to obtain convergence up to order 4 under a mild restriction on the employed step size sequence.

Theorem 4.1. *Let the initial value problem* (3.3) *satisfy Assumptions* C.1 *and* C.2. *Consider for its numerical solution an explicit exponential Rosenbrock method* (2.3) *that fulfills the order conditions of Table* 3.1 *up to order p for some $2 \le p \le 4$. Further, let the step size sequence h_j satisfy the condition*

$$\sum_{k=1}^{n-1}\sum_{j=0}^{k-1} h_j^{p+1} \le C_{\mathtt{H}} \tag{4.1}$$

with a constant $C_{\mathtt{H}}$ that is uniform in $t_0 \le t_n \le T$. Then, for $C_{\mathtt{H}}$ sufficiently small, the numerical method converges with order p. In particular, the numerical solution satisfies the error bound

$$\|u_n - u(t_n)\| \le C\sum_{j=0}^{n-1} h_j^{p+1} \tag{4.2}$$

uniformly on $t_0 \le t_n \le T$. The constant C is independent of the chosen step size sequence satisfying (4.1)

Proof. From (3.16b) we obtain the error recursion

$$e_{n+1} = \mathrm{e}^{h_nJ_n}e_n + h_n\varrho_n - \delta_{n+1}, \qquad e_0 = 0, \tag{4.3}$$

with

$$\varrho_n = \sum_{i=1}^s b_i(h_nJ_n)\Big(g_n(U_{ni}) - G_n(t_n + c_ih_n)\Big).$$

Solving this recursion and using $e_0 = 0$ yields

$$e_n = \sum_{j=0}^{n-1} h_j \, \mathrm{e}^{h_{n-1}J_{n-1}} \cdots \mathrm{e}^{h_{j+1}J_{j+1}} \big(\varrho_j - h_j^{-1}\delta_{j+1}\big). \tag{4.4}$$

Employing lemmas 3.1, 3.2, and 3.3, we obtain the bound

$$\|\varrho_j\| + h_j^{-1}\|\delta_{j+1}\| \le C\big(h_j \,\|e_j\| + \|e_j\|^2 + h_j^p\big). \tag{4.5}$$

Inserting this into (4.4) and using the stability estimate (3.24) yields

$$\|e_n\| \le C \sum_{j=0}^{n-1} h_j \Big(\|e_j\|^2 + h_j \,\|e_j\| + h_j^p\Big). \tag{4.6}$$

The constant in this estimate is uniform as long as

$$\sum_{j=1}^{n-1} \|e_j\| \le C_{\mathtt{A}} \tag{4.7}$$

uniformly holds on $t_0 \le t_n \le T$. The application of a discrete Gronwall lemma to (4.6) then shows the desired bound (4.2).

It still remains to verify that condition (4.7) holds with a uniform bound $C_{\mathtt{A}}$. This follows now recursively from (4.2) and our assumption on the step size sequence (4.1) with $C_{\mathtt{H}}$ sufficiently small. □

In the remainder of this section, we discuss the encountered restriction (4.1) on the step size sequence. For *constant* step sizes, this condition evidently holds with

$$C_{\mathtt{H}} = \frac{1}{2} h^{p-1} (t_n - t_0)^2.$$

Since $p \ge 2$, the size of $C_{\mathtt{H}}$ tends to zero for $h \to 0$.

A similar bound holds for *quasi-uniform* step size sequences where the ratio between the maximal and minimal step length is uniformly bounded. For sequences with increasing step sizes, condition (4.1) is fulfilled as well.

In practice, a problem with (4.1) might occur if the step size suddenly drops by several orders of magnitude. In that case, however, it is possible to modify the above stability analysis and to relax the condition on the step sizes. We briefly explain the idea, but we do not work out all details. If the error at time t_j, say, is large compared to the actual step length, one should rather compare the numerical solution with a smooth trajectory that passes close to u_j. Although u_j might be a non-smooth initial value, such trajectories exist. Then the previous stability proof can be applied once more, at the possible price of increasing the constant C in (3.23) and thus the constants $C_{\mathtt{L}}$ and Ω. As long as this is done only a fixed number of times, stability in (3.24) is still guaranteed.

5. Methods of order up to four. The well known exponential Rosenbrock–Euler method is given by

$$\begin{aligned} u_{n+1} &= \mathrm{e}^{h_n J_n} u_n + h_n \varphi_1(h_n J_n) g_n(u_n) \\ &= u_n + h_n \varphi_1(h_n J_n) F(u_n). \end{aligned} \tag{5.1}$$

It is computationally attractive since it requires only one matrix function per step. The method obviously satisfies Condition 1 of Table 3.1, while Condition 2 is void. Therefore, it is second-order convergent for problems satisfying our analytic framework. A possible error estimator for (5.1) is described in [3].

From the order conditions of Table 3.1, it is straightforward to construct pairs of embedded methods of order 3 and 4. For our variable step size implementation, we consider (2.3b) together with an embedded approximation

$$\widehat{u}_{n+1} = \mathrm{e}^{h_n J_n} u_n + h \sum_{i=1}^{s} \widehat{b}_i(hJ_n)\, g_n(U_{ni}) \tag{5.2}$$

which relies on the same stages U_{ni}. The methods given below were first introduced in [15]. They will be used in the numerical experiments in Section 7.

The scheme `exprb32` consists of a third-order exponential Rosenbrock method with a second-order error estimator (the exponential Rosenbrock–Euler method). Its coefficients are

$$\begin{array}{c|cc} c_1 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \\ & \widehat{b}_1 & \end{array} \quad = \quad \begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1 - 2\varphi_3 & 2\varphi_3 \\ & \varphi_1 & \end{array}$$

The scheme `exprb43` is a fourth-order method with a third-order error estimator. Its coefficients are

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \\ & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 \end{array} \quad = \quad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\,\cdot) & & \\ 1 & 0 & \varphi_1 & \\ \hline & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\ & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3 \end{array}$$

Note that the internal stages of the above methods are just exponential Rosenbrock–Euler steps. This leads to simple methods that can cheaply be implemented.

Evidently, the order conditions of Table 3.1 imply that the weights of any third-order method have to depend on φ_3, whereas that of any fourth-order method depend on φ_3 and φ_4 (in addition to φ_1).

6. Non-autonomous problems. The proposed method can easily be extended to non-autonomous problems

$$u' = F(t,u), \qquad u(t_0) = u_0 \tag{6.1}$$

by rewriting the problem in autonomous form

$$U' = \mathcal{F}(U)\ , \quad U = \begin{bmatrix} t \\ u \end{bmatrix}, \quad \mathcal{F}(U) = \begin{bmatrix} 1 \\ F(t,u) \end{bmatrix} \tag{6.2a}$$

with Jacobian

$$\mathcal{J}_n = \begin{bmatrix} 0 & 0 \\ v_n & J_n \end{bmatrix}, \quad v_n = \frac{\partial}{\partial t} F(t_n, u_n), \quad J_n = \frac{\partial}{\partial u} F(t_n, u_n). \tag{6.2b}$$

This transformation is standard for Rosenbrock methods as well, see [10], but it changes a linear non-autonomous problem into a nonlinear one.

In order to apply our method to the autonomous system (6.2), we have to compute the matrix functions of $\mathcal{J}_n$. Using Cauchy's integral formula and exploiting the special structure of $\mathcal{J}$, we get

$$\varphi(h\mathcal{J}) = \begin{bmatrix} \varphi(0) & 0 \\ h\widehat{\varphi}(hJ)v & \varphi(hJ) \end{bmatrix}, \quad \widehat{\varphi}(z) = \frac{\varphi(z) - \varphi(0)}{z}.$$

For the particular functions in our method, we obtain from (2.5) the relation

$$\widehat{\varphi}_i(hJ) = \varphi_{i+1}(hJ). \tag{6.3}$$

In our formulation, we will work again with the smaller quantities

$$D_{nj} = g_n(t_n + c_j h_n, U_{nj}) - g_n(t_n, u_n) \tag{6.4}$$

where

$$g_n(t,u) = F(t,u) - J_n u - v_n t.$$

Applying method (2.7) to the autonomous formulation (6.2), we get

$$\begin{aligned} U_{ni} &= u_n + h_n c_i \varphi_1(c_i h_n J_n) F(t_n, u_n) \\ &\qquad + h_n^2 c_i^2 \varphi_2(c_i h_n J_n) v_n + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}, \end{aligned} \tag{6.5a}$$

$$u_{n+1} = u_n + h_n \varphi_1(h_n J_n) F(t_n, u_n) + h_n^2 \varphi_2(h_n J_n) v_n + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni}. \tag{6.5b}$$

This is the format of an exponential Rosenbrock method for non-autonomous problems (6.1).

7. Numerical Experiments. We have implemented the exponential Rosenbrock methods `exprb32` and `exprb43` in MATLAB with adaptive time stepping. We employ a standard step size selection strategy based on the local error [10, pp. 28–31]. The error is estimated with the help of the corresponding embedded method from Section 5. Our implementation involves two different options for dealing with the matrix φ-functions: for small examples, we employ diagonalization or Padé approximation for the explicit computation of the matrix functions. For large problems, Krylov subspace methods are used for approximating the product of the matrix functions with the corresponding vectors. For autonomous problems, we use the reformulation (2.7), which requires one Krylov subspace with the vector $F(u_n)$ and $s-1$ Krylov subspaces with the vectors D_{ni}, $i = 2, \ldots, s$. Due to $\|D_{ni}\| = \mathcal{O}(h_n^2)$, these approximations can be computed in very low dimensional subspaces. For non-autonomous problems, the format (6.5) requires one additional Krylov subspace with the vector v_n. Since the term involving v_n is multiplied with h_n^2 (compared to h_n for the other vectors), this subspace will be low dimensional, as well.

EXAMPLE 7.1. As a first example we consider a two-dimensional advection-diffusion-reaction equation for $u = u(x, y, t)$

$$\partial_t u = \varepsilon(\partial_{xx} u + \partial_{yy} u) - \alpha(u_x + u_y) + \gamma u\big(u - \tfrac{1}{2}\big)(1 - u), \qquad (x, y) \in (0,1)^2 \tag{7.1}$$

with homogeneous Neumann boundary conditions and the initial value

$$u(x, y, 0) = 256\big((1-x)x(1-y)y\big)^2 + 0.3,$$

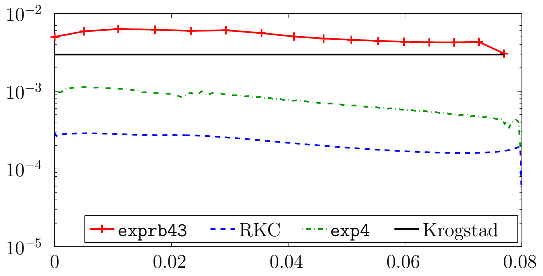

FIG. 7.1. *Step sizes for the advection-diffusion-reaction equation* (7.1) *for* $t \in [0, 0.08]$

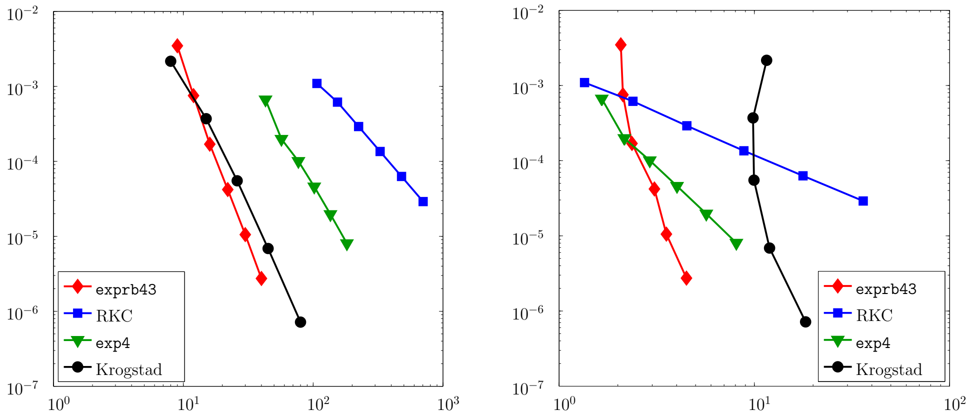

FIG. 7.2. *Number of time steps versus accuracy (left) and CPU time versus accuracy (right) for the advection-diffusion-reaction example* (7.1) *for* $t = 0.08$

where $\varepsilon = 1/100$, $\alpha = -10$, and $\gamma = 100$. The spatial discretization was done with finite differences using 101 grid points in each direction.

This example is taken from [3], where FORTRAN implementations of `exprb43`, combined with the real Leja point method [4], and of the Runge–Kutta–Chebyshev method `RKC` from [25] were compared. Here we compare MATLAB implementations of `RKC`, `exprb43`, `exp4` from [12], and Krogstad's method [17]. The latter three make use of Krylov subspace approximations. To improve the efficiency of the Krogstad method, we reused information from previously computed Krylov subspaces, an approach proposed in [13]. Since an adaptive step-size control based on embedding is not possible for Krogstad's method, we ran this method with constant step size. For this particular example, the step-size control of the other schemes also lead to almost constant steps sizes, see Fig. 7.1. All simulations achieved a final accuracy of about 0.004 at $t = 0.08$. It can be seen that, due to the large advection part, the exponential methods can take much larger steps than `RKC` with `exprb43` taking the largest ones. In total, `exprb43` takes only 18 steps, Krogstad's method takes 27 steps, `exp4` takes 119 steps, while `RKC` uses 383 steps.

In Fig. 7.2, we compare the performance of the Krylov implementations of `exp4`, `exprb43` and Krogstad's method with a MATLAB implementation of `RKC`. Our implementation of `RKC` is based on the well established FORTRAN code by Sommeijer available from the `netlib` repository. Our implementations of `exp4` and `exprb43` allow a maximum dimension of the Krylov subspaces of 36, which is the default

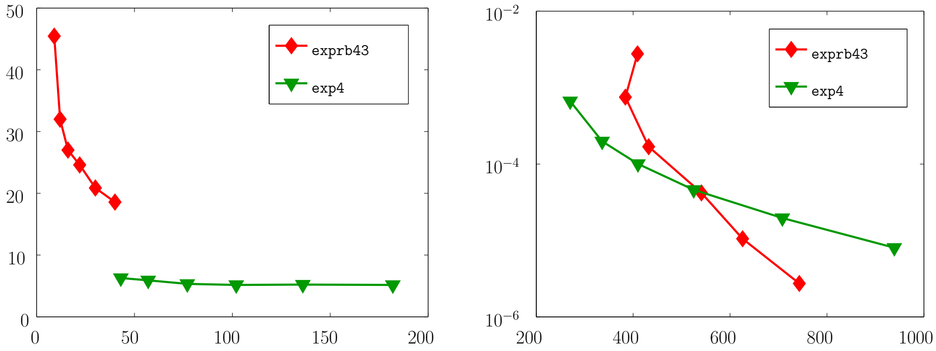

FIG. 7.3. *Number of time steps versus average number of Krylov steps (left) and number of Krylov steps versus accuracy (right) for the advection-diffusion-reaction example* (7.1) *for* $t = 0.08$

value suggested in [12]. The codes were run with tolerances ATOL = RTOL = $10^{-4}, 10^{-4.5}, \ldots, 10^{-6.5}$ (except for Krogstad's method, which was used with constant step size). In the left diagram, we plot the achieved accuracy as a function of the required number of steps. It turns out that, for a given accuracy, the exponential Rosenbrock method `exprb43` uses significantly larger time steps than `exp4` and `RKC`. The number of time steps required for Krogstad's method is about the same as for `exprb43`.

However, the efficiency of a code should also take the cost per time step into account. Therefore, we next consider the CPU time required to achieve a certain accuracy. We are fully aware of the fact that comparing CPU times strongly depends on the available computer architecture, the implementation, and the programming language. Nevertheless, we think that MATLAB comparisons might be of interest.

In Fig. 7.2 we show the achieved accuracy as a function of the required CPU time. It can be seen that for moderate tolerances, `exp4` is faster than `exprb43` while for more stringent tolerances, `exprb43` requires less CPU time. This can be explained by considering the number of Krylov steps used by these methods. In the left diagram in Fig. 7.3 we plotted the average number of Krylov steps over the total number of time steps. Since `exprb43` uses significantly larger time steps, we know from the convergence analysis of Krylov subspace methods [7, 11] that this requires more Krylov steps. The right diagram of Fig. 7.3 shows the achieved accuracy versus the total number of Krylov steps. Since the Krylov approximations dominate the computational cost, this explains the right diagram of Fig. 7.2. Note that, it is impossible to give a reformulation of Krogstad's method in such a way that only one expensive Krylov subspace is required in each step. The gain achieved by reusing previously computed Krylov subspaces [13] does not compensate this disadvantage. Moreover, Krogstad's method has four stages and uses even more matrix functions than `exprb43`.

EXAMPLE 7.2. As a second example, we consider the one-dimensional Schrödinger equation [12] for $\psi = \psi(x,t)$

$$\mathrm{i}\,\frac{\partial \psi}{\partial t} = H(x,t)\psi \tag{7.2a}$$

with the time-dependent Hamiltonian

$$H(x,t) = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \kappa\frac{x^2}{2} + \mu(\sin t)^2\, x\ . \tag{7.2b}$$

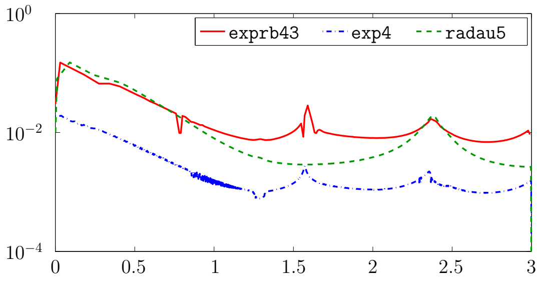

FIG. 7.4. *Step sizes taken by* `exp4`*,* `radau5`*, and* `exprb43` *for the laser example* (7.2) *for* $t \in [0, 3]$

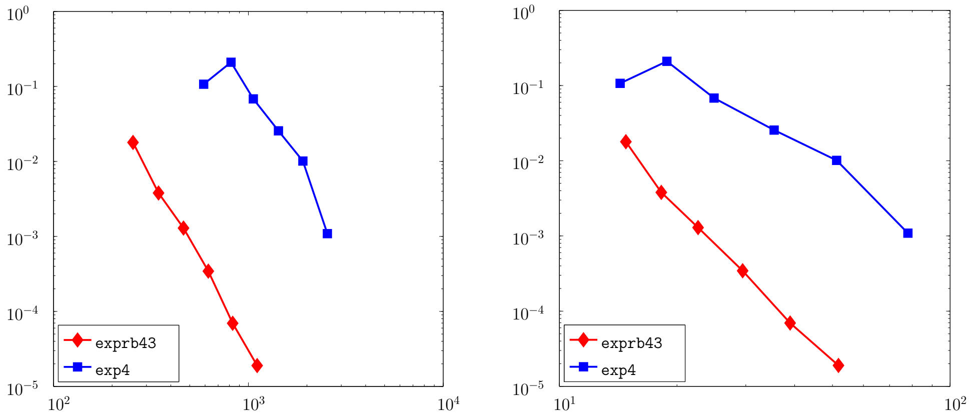

FIG. 7.5. *Number of time steps versus accuracy (left) and CPU time versus accuracy (right) for the laser example* (7.2) *for* $t = 3$

We used the parameter values $\kappa = 10$ and $\mu = 100$. The initial value was chosen as $\psi(x, 0) = \mathrm{e}^{-\sqrt{\kappa}x^2/2}$, which corresponds to the ground state of the unforced harmonic oscillator. Semi-discretization in space was done by a pseudospectral method with 512 Fourier modes on the interval $[-10, 10]$ with periodic boundary conditions.

It was shown in [12] that the MATLAB implementation of `exp4` outperforms MATLAB's standard nonstiff `ode45` method and matrix-free implementations of the stiff solvers `radau5` and `ode15s`. We refer to [12] for details. Here, we use exactly the same spatial discretization but run the simulation until $t = 3$.

In Fig. 7.4, we display the step sizes chosen by the adaptive step-size control for `exp4`, `radau5`, and `exprb43`. The tolerances were set in such a way that all methods achieved a final accuracy of about 0.05. As illustrated in Fig. 7.4, `exprb43` advances with larger step sizes than the other two methods. In total `exprb43` uses 256 steps, `exp4` uses 1906 steps, and `radau5` uses 537 steps. In our implementation of `radau5`, the linear systems arising within the Newton iteration are solved directly while `exp4` and `exprb43` are used with Krylov subspace approximations. The direct solution of the linear systems arising in the `radau5` code result in a total cpu time which is more than 10 times longer than `exprb43`. Since it has been shown in [12] that a much more efficient W-version of `radau5` was still slower than `exp4`, we did not include `radau5` into our run time comparisons.

In Fig. 7.5, we compare the performance of the Krylov implementations of `exp4` and `exprb43`. Both codes were run with tolerances ATOL = RTOL = 10^{-4}, $10^{-4.5}$,

$\dots, 10^{-6.5}$. The diagrams show that the exponential Rosenbrock method `exprb43` uses significantly larger step sizes than `exp4`. Moreover, it is also much faster in terms of total CPU time.

8. Concluding remarks. In this paper we have analyzed the convergence properties of exponential Rosenbrock-type methods in an abstract framework of C_0 semigroups. A local error analysis revealed the stiff order conditions which in turn enabled us to construct methods of orders three and four with embedded error estimates of orders two and three, respectively. To control the error propagation, we derived stability bounds for variable step sizes. This enabled us to give a variable step size convergence proof. We implemented the methods in MATLAB, using Krylov subspace methods to approximate the applications of matrix functions to vectors. The numerical results clearly demonstrate the efficiency of the new integrators.

Appendix A. Analytic semigroups.

So far we restricted our attention to strongly continuous semigroups. This framework, however, limits the class of possible nonlinearities due to Assumption C.2. If the semigroup is even analytic, we can allow more general nonlinearities. In this appendix we sketch how to extend our analysis to this case. For the theoretical background of analytic semigroups, we refer to [8, 24].

ASSUMPTION A.1. *The linear operator A in* (3.3) *is the generator of an analytic semigroup.*

Without loss of generality, we can assume that A is invertible (otherwise we shift it by an appropriate multiple of the identity). Therefore, fractional powers of A are well defined. We choose $0 \le \alpha < 1$ and define $V = \mathcal{D}(A^\alpha) \subset X$. The linear space V is a Banach space with norm $\|v\|_V = \|A^\alpha v\|$.

Our basic assumptions on f are the following:

ASSUMPTION A.2. *We suppose that* (3.3) *possesses a sufficiently smooth solution $u : [0,T] \to V$ with derivatives in V, and that $f : V \to X$ is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are supposed to be uniformly bounded.*

A consequence of Assumption A.1 is that there exist constants C and ω such that

$$\|\mathrm{e}^{tJ}\|_{V \leftarrow V} + \|t^\alpha \mathrm{e}^{tJ}\|_{V \leftarrow X} \le C\,\mathrm{e}^{\omega t}, \qquad t \ge 0 \tag{A.1}$$

holds in a neighborhood of the exact solution.

With these assumptions at hand, we derive once more the bounds of Section 3. Instead of (3.11), we now get

$$\|\Delta_{ni}\|_X + h_n^\alpha \|\Delta_{ni}\|_V \le C h_n^2 \|e_n\|_V + C h_n^3, \tag{A.2}$$

and (3.15) is replaced by

$$\|\delta_{n+1}\|_X + h_n^\alpha \|\delta_{n+1}\|_V \le C h_n^2 \|e_n\|_V + C h_n^{p+1}. \tag{A.3}$$

The same arguments as in the proofs of Lemma 3.2 and 3.3 show the following refined estimates.

LEMMA A.1. *Under Assumptions* A.1 *and* A.2, *we have*

$$\|g_n(U_{ni}) - G_n(t_n + c_i h_n)\|_X \le C\big(h_n + \|e_n\|_V + \|E_{ni}\|_V\big)\,\|E_{ni}\|_V\,, \tag{A.4a}$$

$$\|g_n(u_n) - G_n(t_n)\|_X \le C\,\|e_n\|_V^2\,, \tag{A.4b}$$

$$\Big\|\frac{\partial g_n}{\partial u}\big(u(t_n)\big)\Big\|_{X \leftarrow V} \le C\,\|e_n\|_V\,, \tag{A.4c}$$

and

$$\|E_{ni}\|_V \le C\,\|e_n\|_V + Ch_n^{3-\alpha}, \tag{A.4d}$$

as long as the errors E_{ni} and e_n remain in a sufficiently small neighborhood of 0. □

Further, Assumption A.2 implies

$$\left\|\widehat{J}_n - \widehat{J}_{n-1}\right\|_{X\leftarrow V} \le Ch_{n-1}, \qquad n \ge 1 \tag{A.5}$$

with a constant C that is independent of h_{n-1}. The same arguments as in the proof of Lemma 3.4 with (3.2) replaced by (A.1) now show that

$$\left\|\mathrm{e}^{t\widehat{J}_n} - \mathrm{e}^{t\widehat{J}_{n-1}}\right\|_{V\leftarrow V} \le C_{\mathrm{L}} h_{n-1}\mathrm{e}^{\widetilde{\omega}t}. \tag{A.6}$$

This implies the desired stability estimate in V. For the convergence proof, we need an additional stability result that reflects the parabolic smoothing.

Lemma A.2. *Let the initial value problem* (3.3) *satisfy Assumptions* A.1 *and* A.2, *and let $\widehat{J}_n = \mathrm{D}F(u(t_n))$. Then, for any $\widetilde{\omega} > \omega$, there exists a constant C independent of h_{n-1} such that*

$$\|\mathrm{e}^{h_n\widehat{J}_n}\cdots\mathrm{e}^{h_0\widehat{J}_0}\|_{V\leftarrow X} \;\le\; C\,\frac{\mathrm{e}^{\Omega(h_0+\ldots+h_n)}}{(h_0+\ldots+h_n)^{\alpha}}, \tag{A.7}$$

with $\Omega = C_{\mathrm{L}} + \widetilde{\omega}$ and C_{L} from (A.6).

Proof. Using the same arguments as in [22, Sec. 5] shows this bound. □

We are now in the position to state the convergence proof for exponential Rosenbrock methods in the framework of analytic semigroups. For notational simplicity, we formulate the result for constant step sizes only.

Theorem A.3. *Let the initial value problem* (3.3) *satisfy Assumptions* A.1 *and* A.2 *and consider for its numerical solution an explicit exponential Rosenbrock method* (2.3) *with constant step size h. Assume that the order conditions of Table* 3.1 *hold up to order p with $p = 2$ or $p = 3$. Then, for h sufficiently small, the numerical method converges with order p. In particular, the numerical solution u_n satisfies the uniform error bound*

$$\|u_n - u(t_n)\|_V \le C\,h^p.$$

The constant C depends on T, but it is independent of n and h for $0 \le nh \le T - t_0$.

Proof. We proceed as in the proof of Theorem 4.1. Due to (A.3) and (A.4), we can bound

$$\|\varrho_n\|_X + h^{-1}\|\delta_{n+1}\|_X \le C\big(h\,\|e_n\|_V + \|e_n\|_V^2 + h^p\big). \tag{A.8}$$

By the stability estimate, we now have

$$\|e_n\|_V \le C\sum_{j=0}^{n-1}\frac{h}{(t_n - t_{j+1})^{\alpha}}\Big(h\,\|e_j\|_V + \|e_j\|_V^2 + h^p\Big).$$

The desired error bound thus follows from the application of a discrete Gronwall lemma with weakly singular kernel. □

Remark. For $p \ge 4$, the analysis is much more delicate. Due to (A.4d), the bound (A.8) now contains a term of the order $h^{4-\alpha}$. Under additional assumptions on f, this order reduction can be avoided. For exponential Runge–Kutta methods, this has been detailed in [14]. We do not elaborate this point here.

REFERENCES

[1] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh, *VODE: a variable-coefficient ODE solver*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1038–1051.

[2] G.D. Byrne, *Pragmatic experiments with Krylov methods in the stiff ODE setting*, in: J.R Cash, I. Gladwell (eds.), *Computational Ordinary Differential Equations*, Clarendon Press, Oxford (1992), pp. 323–356.

[3] M. Caliari and A. Ostermann, *Implementation of exponential Rosenbrock-type integrators*, Appl. Numer. Math., (to appear).

[4] M. Caliari, M. Vianello, and L. Bergamaschi, *Interpolating discrete advection-diffusion propagators at Leja sequences*, J. Comput. Appl. Math., 172 (2004), pp. 79–99.

[5] M.P. Calvo and C. Palencia, *A class of explicit multistep exponential integrators for semilinear problems*. Numer. Math., 102 (2006), pp. 367–381.

[6] S.M. Cox and P.C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys., 176 (2002), pp. 430–455.

[7] V.L. Druskin and L.A. Knizhnerman, *Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic*, Numer. Lin. Alg. Appl., 2 (1995), pp. 205–217.

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

[9] A. Friedli, *Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme.* In: R. Bulirsch, R.D. Grigorieff, J. Schröder (Eds.), *Numerical Treatment of Differential Equations*, Lecture Notes in Mathematics 631, Springer, Berlin, 1978.

[10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, 2nd rev. ed., Springer, 1996.

[11] M. Hochbruck and Ch. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[12] M. Hochbruck, Ch. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.

[13] M. Hochbruck and J. Niehoff, *Approximation of matrix operators applied to multiple vectors*, J. Math. Comput. Simul., (to appear).

[14] M. Hochbruck and A. Ostermann, *Explicit exponential Runge–Kutta methods for semilinear parabolic problems*, SIAM J. Numer. Anal., 43 (2005), pp. 1069–1090.

[15] M. Hochbruck and A. Ostermann, *Exponential integrators of Rosenbrock-type*, Oberwolfach Reports, 3 (2006), pp. 1107–1110.

[16] A.-K. Kassam and L.N. Trefethen, *Fourth-order time stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.

[17] S. Krogstad, *Generalized integrating factor methods for stiff PDEs*, J. Comput. Phys., 203 (2005), pp. 72–88.

[18] J.D. Lawson, *Generalized Runge–Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal., 4 (1967), pp. 372–380.

[19] Ch. Lubich and A. Ostermann, *Linearly implicit time discretization of non-linear parabolic equations*, IMA J. Numer. Anal., 15 (1995), pp. 555–583.

[20] Ch. Lubich and A. Ostermann, *Runge–Kutta approximation of quasi-linear parabolic equations*, Math. Comp., 64 (1995), pp. 601–627.

[21] F. Mazzia (Coordinator), *Test Set for IVP Solvers*, `http://pitogora.dm.uniba.it/~testset`.

[22] A. Ostermann and M. Thalhammer, *Convergence of Runge–Kutta methods for nonlinear parabolic equations*, Appl. Numer. Math., 42 (2002), pp. 367–380.

[23] A. Ostermann, M. Thalhammer, and W. Wright, *A class of explicit exponential general linear methods*, BIT, 46 (2006), pp. 409–431.

[24] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer, New York, 1983.

[25] B. Sommeijer, L. Shampine, and J. Verwer, *RKC: an explicit solver for parabolic PDEs*, J. Comput. Appl. Math., 88 (1998), pp. 315–326.

[26] K. Strehmel and R. Weiner, *B-convergence results for linearly implicit one step methods*, BIT, 27 (1987), pp. 264–281.

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

[28] R. Weiner, B.A. Schmitt, and H. Podhaisky, *ROWMAP – a ROW-code with Krylov techniques for large stiff ODEs*, Appl. Numer. Math., 25 (1997), pp. 303–319.