A short course on exponential integrators

Marlis Hochbruck
Department of Mathematics
Karlsruhe Institute of Technology
Germany
E-mail: marlis.hochbruck@kit.edu

Abstract

This paper contains a short course on the construction, analysis, and implementation of exponential integrators for time dependent partial differential equations. A much more detailed recent review can be found in Hochbruck and Ostermann (2010). Here, we restrict ourselves to one-step methods for autonomous problems.

A basic principle for the construction of exponential integrators is the linearization of a semilinear or a nonlinear evolution equation. We distinguish exponential Runge–Kutta methods, using a fixed linearization and exponential Rosenbrock-type methods, which use a continuous linearization at the current approximation of the solution. We present some of the convergence results and give a proof for the simplest method, the exponential Euler method.

The fact that it is possible to construct explicit exponential integrators which obey error bounds even for abstract evolution equations comes at the price that one has to approximate products of matrix functions with vectors in the spatially discrete case. For an efficient implementation one has to combine the integrator with well-chosen algorithms from numerical linear algebra. We briefly sketch Krylov subspace methods for this task.

1 Construction

In this section we will discuss methods for autonomous evolution equations of the form

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

in a finite time interval $t \in [t_0, T]$. Our main interest is in the case where F is a partial differential operator or its spatial discretization. Note that in the abstract case, F involves an unbounded operator and for the discrete case, the norm of the Jacobian grows with the inverse of the spatial mesh width.

For a recent survey on exponential integrators see Hochbruck and Ostermann (2010).

1.1 Runge–Kutta methods

Before we start to introduce exponential integrators, let us briefly review Runge–Kutta methods for the solution of the ordinary differential equation

$$u'(t) = g\big(u(t)\big), \qquad u(t_0) = u_0. \tag{1.2}$$

The function g is assumed to satisfy a Lipschitz condition with a moderate Lipschitz constant (in contrast to F in the evolution equation (1.1)).

The construction of Runge–Kutta methods relies on the following representation of the exact solution u of (1.2) at time $t_{n+1} = t_n + \tau$,

$$u(t_{n+1}) = u(t_n) + \int_0^\tau g\big(u(t_n + \theta)\big)d\theta, \qquad n = 0, 1, \dots. \tag{1.3}$$

The idea is to approximate the integral on the right-hand side by a quadrature formula defined by nodes $0 \le c_1 < \cdots < c_s \le 1$ and weights $b_1, \dots, b_s$. Assume we are given approximations $u_n \approx u(t_n)$ and

$$U_{ni} \approx u(t_n + c_i\tau).$$

Then we have

$$u(t_{n+1}) \approx u_n + \tau \sum_{i=1}^{s} b_i U'_{ni}, \quad \text{where} \quad U'_{ni} = g\big(U_{ni}\big), \quad i = 1, \dots, s.$$

The approximations U_{ni}, $i = 1, \dots, s$, can be obtained by yet other quadrature formulas via

$$u(t_n + c_i\tau) = u(t_n) + \int_{t_n}^{t_n + c_i\tau} u'(\theta)d\theta \approx u_n + \tau \sum_{j=1}^{s} a_{ij} U'_{nj}.$$

A general s-stage Runge–Kutta method is defined as

$$\begin{aligned} u_{n+1} &= u_n + \tau \sum_{i=1}^{s} b_i U'_{ni}, \\ U'_{ni} &= g(U_{ni}), && i = 1, \dots, s \\ U_{ni} &= u_n + \tau \sum_{j=1}^{s} a_{ij} U'_{nj}, && i = 1, \dots, s \end{aligned} \tag{1.4}$$

and denoted in a so-called Butcher tableau as

$$\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}$$

If $a_{ij} = 0$ for $j \geq i$, then U_{ni}, $i = 1, \dots, s$, can be computed explicitly by just evaluating the function g at already computed approximations. Hence, these methods are called explicit Runge–Kutta methods.

Runge–Kutta methods belong to the class of one-step methods, since they only use the current approximation $u_n \approx u(t_n)$ to construct the new approximation $u_{n+1} \approx u(t_n + \tau)$.

Definition 1.1. The local error of a one-step method for solving the initial value problem (1.2) is defined as

$$u_1 - u(t_0 + \tau),$$

where u_1 is the approximation obtained from $u_0 = u(t_0)$ after one step with step size τ.

An important property of methods for solving initial value problems is the order.

Definition 1.2. A numerical scheme for solving the initial value problem (1.2) is of **order** p if for any $g \in C^{p+1}$ the local error is of size $\mathcal{O}(\tau^{p+1})$. If $p \geq 1$, the method is called **consistent**.

Whenever using the notation $\mathcal{O}$, it is of utmost importance to mention all assumptions and quantities which enter the constant defining the set. This is particularly true in the analysis of time-dependent partial differential equations, no matter if the abstract problem is considered in a function space or if a finite difference, a finite element, or a spectral method is used for the spatial discretization.

To verify that a Runge–Kutta method is of a certain order, one has to compute the Taylor expansion of the numerical solution and the Taylor expansion of the exact solution. This requires assumptions on the smoothness of the exact solution (namely that all derivatives used within the expansion are bounded in a suitable norm).

Example 1.3. The simplest method is the well-known explicit Euler method

$$u_{n+1} = u_n + \tau g(u_n), \qquad \text{Butcher tableau} \qquad \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

which is of order one. ◇

1.2 Exponential Runge–Kutta methods

Next we study the construction of exponential Runge–Kutta methods. Such methods are based on linearizing F in (1.1). There are two main options: the first one uses a fixed linearization

$$F\big(u(t)\big) = -Au(t) + g\big(u(t)\big), \qquad A \approx -\frac{\partial F}{\partial u}(u_0). \tag{1.5}$$

The second option is based on a continuous linearization around the current approximation $u_n \approx u(t_n)$

$$\begin{aligned} F\big(u(t)\big) &= -A_n u(t) + g_n\big(u(t)\big), \\ A_n &= -\frac{\partial F}{\partial u}(u_n), \end{aligned} \qquad n = 0, 1, \dots. \tag{1.6}$$

We start with one-step methods based on the fixed linearization (1.5) and consider the initial value problem in a finite dimensional space (say $\mathbb{R}^d$ or $\mathbb{C}^d$). The construction of *exponential Runge–Kutta methods* is closely related to the construction of standard Runge–Kutta methods. It relies on the variation-of-constants formula

$$u(t_n + \tau) = \mathrm{e}^{-\tau A} u(t_n) + \int_0^\tau \mathrm{e}^{-(\tau-\sigma)A} g\big(u(t_n + \sigma)\big)\, d\sigma. \tag{1.7}$$

Here, we used the notation $\mathrm{e}^{-\tau A}$ for the matrix exponential or, in a functional analytic framework, for a semigroup generated by $-A$.

We choose nodes $0 \le c_1 < \ldots < c_s \le 1$ and assume we are given approximations

$$U_{ni} \approx u(t_n + c_i\tau), \qquad i = 1, \dots, s. \tag{1.8}$$

The integral in (1.7) is approximated by a quadrature formula, in which only the nonlinearity g is approximated but the exponential operator (semigroup) is treated exactly:

$$u_{n+1} = \mathrm{e}^{-\tau A} u_n + \tau \sum_{i=1}^{s} b_i(-\tau A) G_{ni}, \tag{1.9a}$$

$$G_{ni} = g\big(U_{ni}\big), \tag{1.9b}$$

$$U_{ni} = \mathrm{e}^{-c_i\tau A} u_n + \tau \sum_{j=1}^{s} a_{ij}(-\tau A) G_{nj}. \tag{1.9c}$$

Note that choosing $A = 0$, *i.e.*, $F = g$, the method reduces to a standard Runge–Kutta method (1.4) with coefficients $b_i(0)$, $a_{ij}(0)$. This Runge–Kutta method will be called the *underlying Runge–Kutta method*. We can gather the exponential Runge–Kutta method in the following tableau

$$\begin{array}{c|c} c_i & a_{ij}(z) \\ \hline & b_j(z) \end{array}, \tag{1.10}$$

where the coefficients are analytic functions which are evaluated at the linear operator $-\tau A$.

Since we use the exponential function within the integrator, a natural requirement is to enforce the integrator to solve linear problems (1.1) with *constant* g exactly. Then

$$u(t_n + \theta\tau) = \mathrm{e}^{-\theta\tau A} u(t_n) + \theta\tau\varphi_1(-\theta\tau A)g, \tag{1.11}$$

where

$$\varphi_1(z) = \int_0^1 \mathrm{e}^{(1-\sigma)z} d\sigma = \frac{\mathrm{e}^z - 1}{z}. \tag{1.12}$$

We use the representation (1.11) for $\theta = c_i$, $i = 1, \ldots, s$ and $\theta = 1$. The conditions

$$u_n = u(t_n), \quad U_{ni} = u(t_n + c_i\tau), \quad n = 0, 1, \ldots, \ i = 1, \ldots, s \tag{1.13}$$

are fulfilled if the *simplifying assumptions*

$$\sum_{i=1}^{s} b_i(z) = \varphi_1(z), \tag{1.14a}$$

$$\sum_{j=1}^{s} a_{ij}(z) = c_i\varphi_1(c_i z), \qquad i = 1, \ldots, s, \tag{1.14b}$$

are satisfied. We restrict ourselves to explicit methods, where $a_{ij}(z) = 0$ for $i \le j$.

Remark. For $z = 0$, the simplifying assumptions just give

$$\sum_{i=1}^{s} b_i(0) = 1, \qquad \sum_{j=1}^{s} a_{ij}(0) = c_i,$$

which ensures that the underlying Runge–Kutta method has stage order $q \ge 1$ (*i.e.*, all inner quadrature formulas are of order at least one) and order $p \ge 1$.

Example 1.4. For $s = 1$, (1.14) yields $b_1(z) = \varphi_1(z)$. The resulting method is called *exponential Euler method*. By $\mathrm{e}^{-z} = 1 - z\varphi_1(-z)$ we have the following equivalent representations of this scheme

$$\begin{aligned} u_{n+1} &= \mathrm{e}^{-\tau A} u_n + \tau\varphi_1(-\tau A) g(t_n, u_n) \\ &= (I - \tau A\varphi_1(-\tau A)) u_n + \tau\varphi_1(-\tau A) g(t_n, u_n) \\ &= u_n + \tau\varphi_1(-\tau A) F(t_n, u_n). \end{aligned} \tag{1.15}$$

For a practical application, where A is a large scale matrix stemming from the space discretization of a differential operator, the latter formula is computationally more efficient, since it requires the evaluation of one product of a matrix function $\varphi_1(-\tau A)$ times a vector, while the first formula requires two such evaluations. ◇

The computation of one time step of the general scheme (1.9) with $s > 1$ requires evaluations of the product of matrix functions with $s + 1$ different vectors, namely with u_n and G_{ni}, $i = 1, \dots, s$. An attractive option to evaluate these products is using an iterative method (for instance a Krylov subspace method with respect to the matrix A and each of the $s + 1$ vectors). We will see later, that it is advantageous to apply the iterative method to vectors of small norm. This can be achieved by defining

$$D_{ni} = G_{ni} - g(u_n), \qquad i = 1, \dots, s. \tag{1.16}$$

Note that for explicit methods, $D_{n1} = 0$, and $\|D_{ni}\| = \mathcal{O}(\tau)$, $i = 2, \dots, s$.

Using the simplifying assumptions (1.14), we can reformulate the method (1.9) equivalently as

$$U_{ni} = u_n + c_i\tau\varphi_1(-c_i\tau A) F(u_n) + \tau \sum_{j=2}^{i-1} a_{ij}(-\tau A) D_{nj}, \tag{1.17a}$$

$$D_{ni} = g(U_{ni}) - g(u_n), \tag{1.17b}$$

$$u_{n+1} = u_n + \tau\varphi_1(-\tau A) F(u_n) + \tau \sum_{i=2}^{s} b_i(-\tau A) D_{ni}. \tag{1.17c}$$

Note that the inner stages U_{ni} and u_{n+1} can be interpreted as corrected exponential Euler approximations with step size $c_i\tau$ and τ, respectively.

The implementation of this method requires the evaluation of the product of matrix functions with s vectors $F(u_n)$, and D_{nj}, $j = 2, \dots, s$. Only the vector $F(u_n)$ is of norm $\mathcal{O}(1)$, but the remaining $s - 1$ vectors D_{nj} are of size $\mathcal{O}(\tau)$, so we can hope that only one expensive approximation is required.

A Matlab software package with an implementation of exponential Runge–Kutta methods and exponential multistep methods for test problems is provided by Berland, Skaflestad, and Wright (2007).

1.3 Exponential Rosenbrock-type methods

These methods are based on the continuous linearization (1.6) of the differential equation (1.1). The idea is similar to that of exponential Runge–Kutta methods, namely we start from the variation-of-constants formula and approximate only g_n but not the exponential operators. We restrict ourselves to the more interesting case of explicit methods satisfying the simplifying assumptions (1.14).

The variation-of-constants formula (1.7) for the solution of

$$u' = F(u) = -A_n u(t) + g_n\big(u(t)\big)$$

yields

$$u(t_n + \theta\tau) = \mathrm{e}^{-\theta\tau A_n} u(t_n) + \int_0^{\theta\tau} \mathrm{e}^{-(\theta\tau-\sigma)A_n} g_n\big(u(t_n+\sigma)\big)\mathrm{d}\sigma.$$

We use this representation for $\theta = c_i$, $i = 1, \dots, s$, and $\theta = 1$. With

$$D_{ni} = g_n(U_{ni}) - g_n(u_n), \tag{1.18a}$$

we define approximations $U_{ni} \approx u(t_n + c_i\tau)$ by

$$\begin{aligned} U_{ni} &= \mathrm{e}^{-c_i\tau A_n} u_n + c_i\tau\varphi_1(-c_i\tau A_n) g_n(u_n) + \tau \sum_{j=2}^{i-1} a_{ij}(-\tau A_n) D_{nj} \\ &= u_n + c_i\tau\varphi_1(-c_i\tau A_n) F(u_n) + \tau \sum_{j=2}^{i-1} a_{ij}(-\tau A_n) D_{nj}, \end{aligned} \tag{1.18b}$$

and

$$u_{n+1} = u_n + \tau\varphi_1(-\tau A_n) F(u_n) + \tau \sum_{i=2}^{s} b_i(-\tau A_n) D_{ni}. \tag{1.18c}$$

Example 1.5. Since $c_1 = 0$, for $s = 1$ we obtain the exponential Rosenbrock–Euler method proposed by Pope (1963).

$$\begin{aligned} u_{n+1} &= \mathrm{e}^{-\tau A_n} u_n + \tau\varphi_1(-\tau A_n) g_n(u_n) \\ &= u_n + \tau\varphi_1(-\tau A_n) F(u_n). \end{aligned} \tag{1.19}$$

It requires only one matrix function per step. ◇

Clearly, as for exponential Runge–Kutta methods, the approximations U_{ni} and u_{n+1} in (1.18) can be interpreted as corrections of the exponential Rosenbrock-Euler method.

2 Error bounds

In this section we present some basic ideas on how to derive error bounds for exponential integrators for partial differential equations. We treat abstract evolution equations and their spatial semidiscretization in a uniform functional analytical framework. This leads to bounds which are independent of the spatial mesh width.

2.1 Analytical framework

We study exponential integrators in a framework of semigroups, see Engel and Nagel (2006) for a short course.

Assumption 2.1. *Let X be a Banach space with norm $\|\cdot\|$. We assume that A is a linear operator on X and that $(-A)$ is the infinitesimal generator of a strongly continuous semigroup e^{-tA} on X.*

This assumption implies

$$\|\mathrm{e}^{-tA}\|_{X\leftarrow X} \le C_A\,\mathrm{e}^{-\omega t}, \quad t \ge 0 \tag{2.1}$$

with $C_A \ge 1$, $\omega \in \mathbb{R}$. We will use only this bound for our error analysis.

For this lecture, the situation in the following example is the relevant one and we just stated the assumption in this generality for people familiar with semigroup theory.

Example 2.2. For $X = \mathbb{R}^n$ or $X = \mathbb{C}^n$, the operator A can be represented by an $n \times n$ matrix $\mathbf{A}$ and $\mathrm{e}^{-t\mathbf{A}}$ is just the matrix exponential function. For the spectral norm, the bound (2.1) is satisfied with $C_{\mathbf{A}} = 1$ if the field of values of $\mathbf{A}$ is contained in the complex half-plane

$$\mathbb{C}_\omega := \{z \in \mathbb{C}\colon \operatorname{Re} z \ge \omega\}.$$

In this case, $\omega = -\mu(-\mathbf{A})$, where $\mu(\mathbf{B}) = \lambda_{\max}\left(\frac{1}{2}\left(\mathbf{B} + \mathbf{B}^H\right)\right)$ is the so-called *logarithmic norm* of the matrix $\mathbf{B}$.

As a special case, the bound (2.1) holds with $\omega = 0$ if the field of values $\mathcal{F}(\mathbf{A})$ of $\mathbf{A}$ defined as

$$\mathcal{F}(\mathbf{A}) = \{\mathbf{x}^H\mathbf{A}\mathbf{x},\ \|\mathbf{x}\| = 1\}$$

is contained in the right complex half-plane. In particular, this is true if $\mathbf{A} = \mathbf{A}^H$ is positive semidefinite or $\mathbf{A} = -\mathbf{A}^H$.

If $\mathbf{A}$ is diagonalizable, $\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{\Lambda}$, then (2.1) holds for an arbitrary matrix norm induced by a vector norm with $C_{\mathbf{A}} = \kappa(\mathbf{X}) = \|\mathbf{X}\|\ \left\|\mathbf{X}^{-1}\right\|$ if the spectrum of $\mathbf{A}$ is contained in $\mathbb{C}_\omega$.

The crucial observation is that these assumptions are independent of the dimension of $\mathbf{A}$. This will allow to prove temporal convergence results that are independent of the spatial mesh. ◇

2.2 Exponential Runge–Kutta methods: analysis

Our analysis will make use of the variation-of-constants formula (1.7) for the solution of

$$u'(t) + Au(t) = g\big(u(t)\big), \quad u(t_0) = u_0, \tag{2.2}$$

i.e., (1.1) with fixed linearization (1.5). In order to simplify the notation, we set

$$f(t) = g\big(u(t)\big).$$

For the nonlinearity g we make the following assumption (for an analysis under more general assumptions we refer to Hochbruck and Ostermann (2005)):

Assumption 2.3. *We assume that $g : [0,T] \times X \to X$ is locally Lipschitz-continuous in a strip along the exact solution u. Thus, there exists a real number $L = L(R,T)$ such that, for all $t \in [0,T]$,*

$$\|g(v) - g(w)\| \le L\|v - w\| \tag{2.3}$$

if $\max\big(\|v - u(t)\|, \|w - u(t)\|\big) \le R$.

Our proofs are heavily based on the representation of the exact solution by the variation-of-constants formula (1.7), which coincides with

$$u(t_{n+1}) = \mathrm{e}^{-\tau A} u(t_n) + \int_0^\tau \mathrm{e}^{-(\tau-\sigma)A} f(t_n + \sigma)\, d\sigma. \tag{2.4}$$

in our notation.

In order to analyze exponential Runge–Kutta methods, we expand $f(t)$ in a Taylor series with remainder in integral form and insert it into (2.4):

$$\begin{aligned} u(t_{n+1}) &= \mathrm{e}^{-\tau A} u(t_n) + \int_0^\tau \mathrm{e}^{-(\tau-\sigma)A} f(t_n + \sigma)\, d\sigma \\ &= \mathrm{e}^{-\tau A} u(t_n) + \tau \sum_{k=1}^{p} \varphi_k(-\tau A)\tau^{k-1} f^{(k-1)}(t_n) \\ &\quad + \int_0^\tau \mathrm{e}^{-(\tau-\sigma)A} \int_0^\sigma \frac{(\sigma-\xi)^{p-1}}{(p-1)!} f^{(p)}(t_n + \xi)\, d\xi\, d\sigma. \end{aligned} \tag{2.5}$$

Here we used the φ-functions defined as

$$\varphi_k(z) = \int_0^1 \mathrm{e}^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \ge 1. \tag{2.6}$$

These functions satisfy $\varphi_k(0) = 1/k!$ and the recurrence relation

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_0(z) = \mathrm{e}^z. \tag{2.7}$$

Assumption 2.1 enables us to define the operators

$$\varphi_k(-\tau A) = \int_0^1 \mathrm{e}^{-\tau(1-\theta)A} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1.$$

The following lemma turns out to be crucial.

Lemma 2.4. *Under Assumption 2.1, the operators $\varphi_k(-\tau A)$, $k = 1, 2, \ldots$, are bounded on X.*

Proof. The boundedness simply follows from the estimate

$$\|\varphi_k(-\tau A)\|_{X \leftarrow X} \leq \int_0^1 \|\mathrm{e}^{-\tau(1-\theta)A}\|_{X \leftarrow X} \frac{\theta^{k-1}}{(k-1)!} d\theta$$

and the bound (2.1) on the semigroup. □

We now present an error bound for the simplest exponential integrator, the exponential Euler method.

The Taylor expansion of f with remainder in integral form is given by

$$f(t_n + \sigma) = f(t_n) + \int_0^\sigma f'(t_n + \theta) d\theta. \tag{2.8}$$

Inserting the exact solution into the numerical scheme yields

$$u(t_{n+1}) = \mathrm{e}^{-\tau A} u(t_n) + \tau \varphi_1(-\tau A) f(t_n) + \delta_{n+1}, \tag{2.9}$$

where, by (2.5), the defect is given by

$$\delta_{n+1} = \int_0^\tau \mathrm{e}^{-(\tau-\sigma)A} \int_0^\sigma f'(t_n + \theta) d\theta d\sigma. \tag{2.10}$$

For this defect we have the following estimate.

Lemma 2.5. *Let the semilinear initial value problem satisfy Assumption 2.1 and assume that $f' \in L^\infty(0, T; X)$. Then*

$$\left\| \sum_{j=0}^{n-1} \mathrm{e}^{-j\tau A} \delta_{n-j} \right\| \leq C\tau M, \qquad M := \sup_{0 \leq t \leq t_n} \|f'(t)\| \tag{2.11}$$

holds with a constant $C = C(C_A, \omega, t_n)$, uniformly in $0 \leq t_n \leq T$.

Proof. We write

$$\mathrm{e}^{-j\tau A}\delta_{n-j} = \mathrm{e}^{-j\tau A}\int_0^\tau \mathrm{e}^{-(\tau-\sigma)A}\int_0^\sigma f'(t_{n-j-1}+\theta)d\theta d\sigma.$$

Using the stability bound (2.1), the sum is bounded by

$$\Big\|\sum_{j=0}^{n-1}\mathrm{e}^{-j\tau A}\delta_{n-j}\Big\| \le CM\tau^2 n \le CM\tau t_n.$$

This proves the desired estimate. □

For the exponential Euler method, we have the following convergence result.

Theorem 2.6. *Let the initial value problem (2.2) satisfy Assumption 2.1, and consider for its numerical solution the exponential Euler method (1.15). Further assume that $f : [0,T] \to X$ is differentiable with $f' \in L^\infty(0,T;X)$. Then, the error bound*

$$\|u_n - u(t_n)\| \le C\,\tau \sup_{0\le t\le t_n}\|f'(t)\|$$

holds uniformly in $0 \le n\tau \le T$. The constant C depends on T, but it is independent of n and τ.

Proof. Let the error be denoted by $e_n = u_n - u(t_n)$. Then the exponential Euler method satisfies the error recursion

$$e_{n+1} = \mathrm{e}^{-\tau A}e_n + \tau\varphi_1(-\tau A)\big(g(t_n,u_n) - f(t_n)\big) - \delta_{n+1} \tag{2.12}$$

with defect δ_{n+1} defined in (2.10). Solving this recursion yields

$$e_n = \tau\sum_{j=0}^{n-1}\mathrm{e}^{-(n-j-1)\tau A}\varphi_1(-\tau A)\big(g(t_j,u_j) - f(t_j)\big) - \sum_{j=0}^{n-1}\mathrm{e}^{-j\tau A}\delta_{n-j}.$$

Using (2.1) and Lemma 2.5, we may estimate this by

$$\|e_n\| \le C\tau\sum_{j=0}^{n-1}\|e_j\| + C\tau\sup_{0\le t\le t_n}\|f'(t)\|.$$

The application of the discrete Gronwall Lemma 2.7 concludes the proof. □

In the previous proof we used the following standard discrete Gronwall Lemma.

Lemma 2.7 (Discrete Gronwall Lemma)**.** *For $\tau > 0$ and $T > 0$, let $0 \le t_n = n\tau \le T$. Further assume that the sequence of non-negative numbers ε_n satisfies the inequality*

$$\varepsilon_n \le a\tau \sum_{\nu=1}^{n-1} \varepsilon_\nu + b$$

for some $a, b \ge 0$. Then the estimate $\varepsilon_n \le Cb$ holds, where the constant C depends on a and on T.

The convergence analysis of higher-order methods turns out to be much more complicated than that for the exponential Euler scheme, due to the low order of the internal stages. The order conditions in Table 1 contain the functions

$$\psi_j(-\tau A) = \varphi_j(-\tau A) - \sum_{i=1}^{s} b_i(-\tau A)\frac{c_i^{j-1}}{(j-1)!} \tag{2.13}$$

and

$$\psi_{j,i}(-\tau A) = \varphi_j(-c_i\tau A)c_i^j - \sum_{k=1}^{i-1} a_{ik}(-\tau A)\frac{c_k^{j-1}}{(j-1)!} \tag{2.14}$$

which arise in the Taylor expansion within the variation-of-constants formula.

Theorem 2.8. (Hochbruck and Ostermann (2005))
Let the initial value problem (2.2) satisfy Assumptions 2.9 and 2.10 and consider for its numerical solution an explicit exponential Runge–Kutta method (1.9) satisfying (1.14). For $2 \le p \le 4$, assume that the order conditions of Table 1 hold up to order $p-1$ and that $\psi_p(0) = 0$. Further assume that the remaining conditions of order p hold in a weaker form with $b_i(0)$ instead of $b_i(-\tau A)$ for $2 \le i \le s$. Then the numerical solution u_n satisfies the error bound

$$\|u_n - u(t_n)\| \le C\,\tau^p$$

uniformly in $0 \le n\tau \le T$. The constant C depends on T, but it is independent of n and τ.

Examples of higher order methods and references to them can be found in Hochbruck and Ostermann (2005), Hochbruck and Ostermann (2010), and Luan and Ostermann (2014a).

Table 1: Stiff order conditions for explicit exponential Runge–Kutta methods for $\alpha = 0$. Here J and K denote arbitrary bounded operators on X. The functions ψ_i and $\psi_{k,\ell}$ are defined in (2.13) and (2.14), respectively.

Number	Order	Order condition
1	1	$\psi_1(-\tau A) = 0$
2	2	$\psi_2(-\tau A) = 0$
3	2	$\psi_{1,i}(-\tau A) = 0$
4	3	$\psi_3(-\tau A) = 0$
5	3	$\sum_{i=1}^{s} b_i(-\tau A) J \psi_{2,i}(-\tau A) = 0$
6	4	$\psi_4(-\tau A) = 0$
7	4	$\sum_{i=1}^{s} b_i(-\tau A) J \psi_{3,i}(-\tau A) = 0$
8	4	$\sum_{i=1}^{s} b_i(-\tau A) J \sum_{j=2}^{i-1} a_{ij}(-\tau A) J \psi_{2,j}(-\tau A) = 0$
9	4	$\sum_{i=1}^{s} b_i(-\tau A) c_i K \psi_{2,i}(-\tau A) = 0$

2.3 Exponential Rosenbrock-type methods: analysis

For the error analysis of (1.18), we work in a semigroup framework. Background information on semigroups can be found in the textbooks Engel and Nagel (2000); Pazy (1992). Let

$$J = J(u) = \mathrm{D}F(u) = \frac{\partial F}{\partial u}(u) \tag{2.15}$$

be the Fréchet derivative of F in a neighborhood of the exact solution of (1.5). Throughout the paper we consider the following assumptions.

Assumption 2.9. *The linear operator $J = J(u)$ 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*

$$\left\|\mathrm{e}^{tJ}\right\|_{X \leftarrow X} \leq C\, \mathrm{e}^{\omega t}, \qquad t \geq 0 \tag{2.16}$$

holds uniformly in a neighborhood of the exact solution of (1.5).

In the subsequent analysis, we restrict our attention to autonomous semilinear problems,

$$u'(t) = F\big(u(t)\big), \quad F(u) = -Au + g(u), \quad u(t_0) = u_0. \tag{2.17}$$

This implies that (1.6) takes the form

$$-A_n = -A + \frac{\partial g}{\partial u}(u_n), \quad g_n\big(u(t)\big) = g\big(u(t)\big) - \frac{\partial g}{\partial u}(u_n)u(t). \tag{2.18}$$

We suppose that A satisfies Assumption 2.1. Our main hypothesis on the nonlinearity g is the following:

Assumption 2.10. *We assume that (2.2) possesses a sufficiently smooth solution $u : [0, T] \to X$ with derivatives in X, and that $g : [0, T] \times X \to X$ is sufficiently often Fréchet-differentiable in a strip along the exact solution. All occurring derivatives are assumed to be uniformly bounded.*

The latter assumption implies that the Jacobian (2.15) satisfies the Lipschitz condition

$$\|J(u) - J(v)\|_{X \leftarrow X} \le C \|u - v\| \tag{2.19}$$

in a neighborhood of the exact solution.

Theorem 2.11. (Hochbruck et al., 2009, Theorem 4.1) *Suppose the initial value problem (2.17) satisfies Assumption 2.1 and 2.10. Consider for its numerical solution an explicit exponential Rosenbrock method (1.18) that fulfills the order conditions of Table 2 up to order p for some $2 \le p \le 4$. Further, let the step size sequence τ_j satisfy the condition*

$$\sum_{k=1}^{n-1} \sum_{j=0}^{k-1} \tau_j^{p+1} \le C_\tau \tag{2.20}$$

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

$$\|u_n - u(t_n)\| \le C \sum_{j=0}^{n-1} \tau_j^{p+1} \tag{2.21}$$

uniformly on $t_0 \le t_n \le T$. The constant C is independent of the chosen step size sequence satisfying (2.20).

The well-known exponential Rosenbrock–Euler method (1.19) obviously satisfies condition 1 of Table 2, while condition 2 is void. Therefore, it is second-order convergent for problems satisfying our analytic framework. A possible error estimator for (1.19) is described in Caliari and Ostermann (2009).

Table 2: Stiff order conditions for exponential Rosenbrock methods applied to autonomous problems.

Number	Order condition	Order
1	$\sum_{i=1}^{s} b_i(z) = \varphi_1(z)$	1
2	$\sum_{j=1}^{i-1} a_{ij}(z) = c_i\varphi_1(c_i z), \quad 2 \le i \le s$	2
3	$\sum_{i=2}^{s} b_i(z)c_i^2 = 2\varphi_3(z)$	3
4	$\sum_{i=2}^{s} b_i(z)c_i^3 = 6\varphi_4(z)$	4

Example 2.12. Hochbruck, Lubich, and Selhofer (1998) proposed the following class of exponential integrators

$$
\begin{aligned}
k_i &= \varphi_1(-\gamma\tau A_n)\Big(-A_n u_n + g_n(U_{ni}) - \tau A_n \sum_{j=1}^{i-1} \beta_{ij} k_j\Big),\\
U_{ni} &= u_n + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j, \quad i = 1, \dots, s,\\
u_{n+1} &= u_n + \tau \sum_{i=1}^{s} b_i k_i,
\end{aligned}
$$

where $\gamma, \alpha_{ij}, \beta_{ij}, b_i$ are coefficients that determine the method.

Note that in contrast to general Rosenbrock methods, this method uses the φ_1-function only. It thus cannot have order larger than two.

However, the method `exp4` was designed such that the computation of the matrix functions is particularly efficient. To achieve this, it was not written as a three-stage exponential Rosenbrock-type method but as a seven-stage method, which uses only three function evaluations. The code comes with error and step size control and uses Krylov approximations for the approximation of the matrix functions.

◇

The `exp4` method of Hochbruck, Lubich, and Selhofer (1998) led to a revival of exponential integrators and initiated a lot of activities on the construction, implementation, analysis, and applications of such methods.

Examples of higher order exponential Rosenbrock-type methods and references to these methods can be found in Hochbruck, Ostermann, and Schweitzer (2009) and Hochbruck and Ostermann (2010). A comparative study of the performance of exponential, implicit, and explicit integrators for stiff systems of ordinary differential equations is given in Loffeld and

Tokman (2013). Variants of exponential Runge–Kutta methods which are designed for efficient implementations have been proposed in Tokman (2006, 2011) and Tokman, Loffeld, and Tranquilli (2012). Higher order methods Rosenbrock methods are constructed and analyzed by Luan and Ostermann (2014b)

3 Approximation of the matrix exponential operator

This section deals with the approximation of products of a function of a matrix with a vector,

$$\mathbf{x} = \phi(\mathbf{A})\mathbf{b}, \qquad \mathbf{A} \in \mathbb{C}^{n,n}, \quad \mathbf{b} \in \mathbb{C}^n, \quad \|\mathbf{b}\| = 1, \tag{3.1}$$

where $\phi : \mathbb{C} \to \mathbb{C}$ is analytic in a neighborhood of the spectrum of $\mathbf{A}$ or in some cases even in a neighborhood of the field of values of $\mathbf{A}$.

We refer to the excellent monograph Higham (2008) for a detailed study of matrix functions. A survey on matrix functions with emphasis on the matrix exponential was given by Frommer and Simoncini (2008).

Here, we are interested in $\phi(z) = e^z$ or $\phi(z) = \varphi_k(z)$, $k = 1, 2, \ldots$, which arise in exponential integrators.

3.1 Arnoldi algorithm

Discretizations of partial differential operators lead to matrices $\mathbf{A}$ which are large and sparse. Since, in general, $\mathbf{A}^j$ is no longer sparse if j is large, the same holds for $\phi(\mathbf{A})$. In many applications, it is sufficient to compute $\phi(\mathbf{A})\mathbf{b}$ and this can be accomplished without computing $\phi(\mathbf{A})$ explicitly. Here, we are interested in approximations in certain Krylov subspaces.

There are several possible ways to motivate Krylov approximations to matrix functions. Here, we follow the derivation of Hochbruck and Lubich (1997). Note that Cauchy's integral formula,

$$\phi(\mathbf{A})\mathbf{b} = \frac{1}{2\pi i} \int_\Gamma \phi(\lambda)(\lambda\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\, d\lambda = \frac{1}{2\pi i} \int_\Gamma \phi(\lambda)\mathbf{x}(\lambda)\, d\lambda \tag{3.2}$$

is based on the solution of shifted linear systems

$$(\lambda\mathbf{I} - \mathbf{A})\mathbf{x}(\lambda) = \mathbf{b}, \qquad \lambda \in \Gamma. \tag{3.3}$$

These linear systems can be approximated in a Krylov subspace defined as

$$\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \operatorname{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{m-1}\mathbf{b}\}. \tag{3.4}$$

We restrict ourselves to Arnoldi-based methods, which compute an orthonormal basis $\mathbf{V}_m \in \mathbb{C}^{n,m}$ of $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$. The derivation can be done for Lanczos-based methods in an analogous way. For Hermitian or skew-Hermitian matrices, the Lanczos and the Arnoldi algorithm are equivalent.

The Arnoldi recurrence reads

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^T, \qquad \mathbf{V}_m^H\mathbf{V}_m = \mathbf{I}_m,$$

where

$$\mathbf{V}_m^H \mathbf{A} \mathbf{V}_m = \mathbf{H}_m. \tag{3.5}$$

A simple implementation is given in the following algorithm.

Algorithm 3.1. (Arnoldi algorithm, see, *e.g.*, Saad (1992))

Given $\mathbf{A} \in \mathbb{C}^{n,n}$, $\mathbf{b} \in \mathbb{C}^n$, and $\beta = \|\mathbf{b}\| > 0$
$\mathbf{v}_1 = \mathbf{b}/\beta$
for $m = 1, 2, \ldots$

- for $j = 1, \ldots, m$
  $h_{j,m} = \langle \mathbf{v}_j, \mathbf{A}\mathbf{v}_m \rangle$
- $\widetilde{\mathbf{v}}_{m+1} = \mathbf{A}\mathbf{v}_m - \sum_{j=1}^{m} h_{j,m}\mathbf{v}_j$
- $h_{m+1,m} = \sqrt{\langle \widetilde{\mathbf{v}}_{m+1}, \widetilde{\mathbf{v}}_{m+1} \rangle}$
- $\mathbf{v}_{m+1} = \widetilde{\mathbf{v}}_{m+1}/h_{m+1,m}$

Here, we will use the standard Euclidean inner product for the ease of presentation. However, the inner product within the Arnoldi algorithm should be chosen according to the application. In particular, for finite element disretizations, $\mathbf{V}_m$ should be orthonormal with respect to the discrete L^2 inner product, *i.e.*, $\mathbf{V}_m^H \mathbf{M} \mathbf{V}_m = \mathbf{I}_m$. The algorithm and the results below can be adapted without any difficulty.

The following properties of the Arnoldi algorithm are easily verified.

Lemma 3.2. *Let* $\mathbf{V}_m$ *and* $\mathbf{H}_m$ *be the matrices from the Arnoldi algorithm applied to* $\mathbf{A}$ *and* $\mathbf{b}$ *with* $\|\mathbf{b}\| = 1$. *Then*

(a) $\mathcal{F}(\mathbf{H}_m) \subseteq \mathcal{F}(\mathbf{A})$,

(b) $p_{m-1}(\mathbf{A})\mathbf{b} = \mathbf{V}_m p_{m-1}(\mathbf{H}_m)\mathbf{e}_1$ *for all polynomials of degree at most* $m-1$.

Proof. Exercise. □

The definition of a Krylov subspace implies that

$$\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \mathcal{K}_m(\lambda\mathbf{I} - \mathbf{A}, \mathbf{b}) \qquad \text{for all} \qquad \lambda \in \mathbb{C}.$$

Moreover, the relation

$$(\lambda\mathbf{I} - \mathbf{A})\mathbf{V}_m = \mathbf{V}_m(\lambda\mathbf{I} - \mathbf{H}_m) - h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^T$$

holds.

The Galerkin approximation $\mathbf{x}_m(\lambda) \in \mathcal{K}_m(\mathbf{A}, \mathbf{b})$ for the solution $\mathbf{x}(\lambda)$ of (3.3) is defined by the condition that the residual

$$\mathbf{r}_m(\lambda) = \mathbf{b} - (\lambda\mathbf{I} - \mathbf{A})\mathbf{x}_m(\lambda) \tag{3.6}$$

is orthogonal to $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$. Writing $\mathbf{x}_m(\lambda) = \mathbf{V}_m \mathbf{y}_m(\lambda)$, this is equivalent to

$$0 = \mathbf{V}_m^H \mathbf{r}_m(\lambda) = \mathbf{e}_1 - (\lambda \mathbf{I} - \mathbf{H}_m)\mathbf{y}_m(\lambda).$$

If we choose a curve Γ which includes the field of values $\mathcal{F}(\mathbf{A})$ of $\mathbf{A}$ in its interior, and by Lemma 3.2 also $\mathcal{F}(\mathbf{H}_m)$, then Γ does not contain any eigenvalue of $\mathbf{H}_m$. Hence, $\lambda \mathbf{I} - \mathbf{H}_m$ is nonsingular and

$$\mathbf{x}_m(\lambda) = \mathbf{V}_m(\lambda \mathbf{I} - \mathbf{H}_m)^{-1}\mathbf{e}_1. \tag{3.7}$$

This is the approximation of the full orthogonalization method (FOM) by Saad (1981).

An approximation to $\phi(\mathbf{A})\mathbf{b}$ is obtained by approximating $\mathbf{x}_m(\lambda)$ within the Cauchy integral formula by its FOM iterate:

$$\begin{aligned}
\phi(\mathbf{A})\mathbf{b} &\approx \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)\mathbf{x}_m(\lambda)d\lambda \\
&= \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)\mathbf{V}_m(\lambda \mathbf{I} - \mathbf{H}_m)^{-1}\mathbf{e}_1 d\lambda \\
&= \mathbf{V}_m \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda \mathbf{I} - \mathbf{H}_m)^{-1} d\lambda \, \mathbf{e}_1 \\
&= \mathbf{V}_m \phi(\mathbf{H}_m)\mathbf{e}_1.
\end{aligned} \tag{3.8}$$

The last identity is again a Cauchy integral formula, but this time for the tiny matrix $\mathbf{H}_m$. For such a matrix, $\phi(\mathbf{H}_m)$ can be computed or approximated explicitly by diagonalization of $\mathbf{H}_m$ or by Padé approximation.

Note that we used the solution of the linear systems only for the purpose of deriving the approximation $\mathbf{x}_m$. In practice, we do not have to solve any of these linear systems but instead we just compute $\phi(\mathbf{H}_m)$. However, it is worth mentioning, that one could also choose certain nodes θ_k on the curve Γ and construct a quadrature formula, which then requires the solution of a small number of linear systems to approximate $\mathbf{x}_m(\theta_k)$, see, *e.g.*, Trefethen, Weideman, and Schmelzer (2006) and Hale, Higham, and Trefethen (2008).

3.2 Stopping criteria

The above derivation via the Cauchy integral formula also motivates a stopping criterion for approximations to matrix functions, cf. (Hochbruck and Lubich, 1997, Section 6.3). For linear systems

$$(\lambda \mathbf{I} - \tau \mathbf{A})\mathbf{x}(\lambda) = \mathbf{b}$$

it is usually based on the residual $\mathbf{r}_m(\lambda)$ instead of the error

$$\boldsymbol{\epsilon}(\lambda) = \mathbf{x}_m(\lambda) - \mathbf{x}(\lambda).$$

The residuals for the linear systems satisfy

$$\mathbf{r}_m(\lambda) = \tau h_{m+1,m} \big(\mathbf{e}_m^T (\lambda \mathbf{I} - \tau \mathbf{H}_m)^{-1} \mathbf{e}_1\big) \mathbf{v}_{m+1}.$$

Using Cauchy's integral formula, the error of the mth Krylov approximation to $\phi(-\tau\mathbf{A})$ can be written as

$$\boldsymbol{\epsilon}_m = \mathbf{V}_m \phi(\tau \mathbf{H}_m)\mathbf{e}_1 - \phi(-\tau\mathbf{A})\mathbf{b} = \frac{1}{2\pi i} \int_\Gamma \phi(\lambda) \boldsymbol{\epsilon}_m(\lambda) d\lambda,$$

where Γ is a contour enclosing the eigenvalues of $\tau\mathbf{A}$. Replacing the error $\boldsymbol{\epsilon}_m(\lambda)$ by $\mathbf{r}_m(\lambda)$, we get the *generalized residual*

$$\mathbf{r}_m = \frac{1}{2\pi i} \int_\Gamma \phi(\lambda) \mathbf{r}_m(\lambda) d\lambda = \tau h_{m+1,m} \big(\mathbf{e}_m^T \phi(\tau \mathbf{H}_m)\mathbf{e}_1\big)\, \mathbf{v}_{m+1},$$

which can be computed by no additional cost. The same stopping criterion was proposed earlier by Saad (1992) with a different motivation and in Lubich (2008, p. 94), a refined version can be found for the case of $\phi(z) = e^z$ and skew-Hermitian matrices $\mathbf{A}$. A more detailed discussion on stopping criteria can be found in Botchev, Grimm, and Hochbruck (2013).

In Hochbruck and Lubich (1997) it was shown how to derive error bounds for the approximation of the matrix exponential based on Cauchy's integral formula by using Faber polynomials for different sets in the complex plane. For the special case of skew Hermitian matrices, error bounds and stopping criteria are given in Lubich (2008). Here, the analysis is based on Chebyshev series.

3.3 Software

For the polynomial approximation of the matrix exponential, there is excellent Matlab software available. Here is a list of packages, which are freely available.

- `expmv`: Matlab code by Al-Mohy and Higham (2011). It is based on a Taylor expansion and includes applications to exponential integrators.
- `expmvp` and `phipm`: Matlab code by Niesen and Wright (2012). It contains an implementation of Krylov subspace methods for approximating φ functions

- `expokit`: Matlab and Fortran codes by Sidje (1998). This codes approximates $e^{\tau \mathbf{A}}\mathbf{b}$ and uses a time-stepping procedure, if the number of iterations exceeds a given limit (default value is $m_{\max} = 30$).

Acknowledgement

The author likes to thank Volker Grimm and Tanja Göckler for their help in preparing the lectures for the Gene Golub SIAM summer school in Shanghai 2013 and for their careful reading of this paper.

References

A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM J. Sci. Comp.*, 33(2):488–511, 2011. URL http://link.aip.org/link/?SCE/33/488/1.

H. Berland, B. Skaflestad, and W. M. Wright. EXPINT—A MATLAB package for exponential integrators. *ACM Trans. Math. Softw.*, 33(1), Mar. 2007. URL http://doi.acm.org/10.1145/1206040.1206044.

M. A. Botchev, V. Grimm, and M. Hochbruck. Residual, Restarting, and Richardson Iteration for the Matrix Exponential. *SIAM J. Sci. Comput.*, 35(3):A1376–A1397, 2013. URL http://dx.doi.org/10.1137/110820191.

M. Caliari and A. Ostermann. Implementation of exponential Rosenbrock-type integrators. *Appl. Numer. Math.*, 59(3-4):568–581, 2009. URL http://www.sciencedirect.com/science/article/B6TYD-4S3WWWF-2/2/2812d6299c63c61465401cd7e7085815.

K.-J. Engel and R. Nagel. *One-parameter semigroups for linear evolution equations*, volume 194 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000. With contributions by S. Brendle, M. Campiti, T. Hahn, G. Metafune, G. Nickel, D. Pallara, C. Perazzoli, A. Rhandi, S. Romanelli and R. Schnaubelt.

K.-J. Engel and R. Nagel. *A short course on operator semigroups*. Universitext. Springer, New York, 2006.

A. Frommer and V. Simoncini. Matrix functions. In *Model order reduction: theory, research aspects and applications*, volume 13 of *Math. Ind.*, pages 275–303. Springer, Berlin, 2008. URL http://dx.doi.org/10.1007/978-3-540-78841-6_13.

N. Hale, N. J. Higham, and L. N. Trefethen. Computing $\mathbf{A}^\alpha$, $\log(\mathbf{A})$, and related matrix functions by contour integrals. *SIAM J. Numer. Anal.*, 46(5):2505–2523, 2008. URL http://dx.doi.org/10.1137/070700607.

N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia, 2008.

M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997. URL http://dx.doi.org/10.1137/S0036142995280572.

M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19(5): 1552–1574, 1998. URL http://link.aip.org/link/?SCE/19/1552/1.

M. Hochbruck and A. Ostermann. Explicit exponential Runge–Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.*, 43(3):1069–1090, 2005. URL http://link.aip.org/link/?SNA/43/1069/1.

M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. URL http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=7701740&fulltextType=RA&fileId=S0962492910000048.

M. Hochbruck, A. Ostermann, and J. Schweitzer. Exponential Rosenbrock-type methods. *SIAM J. Numer. Anal.*, 47(1):786–803, 2009. URL http://link.aip.org/link/?SNA/47/786/1.

J. Loffeld and M. Tokman. Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs. *J. Comput. Appl. Math.*, 241:45–67, 2013. URL http://dx.doi.org/10.1016/j.cam.2012.09.038.

V. T. Luan and A. Ostermann. Explicit exponential Runge–Kutta methods of high order for parabolic problems. *J. Comput. Appl. Math.*, 256: 168–179, 2014a. URL http://dx.doi.org/10.1016/j.cam.2013.07.027.

V. T. Luan and A. Ostermann. Exponential Rosenbrock methods of order five — construction, analysis and numerical comparisons. *J. Comput. Appl. Math.*, 255:417–431, 2014b. URL http://dx.doi.org/10.1016/j.cam.2013.04.041.

C. Lubich. *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*. Zurich Lectures in Advanced Mathematics. European Mathematical Society (EMS), 2008.

J. Niesen and W. M. Wright. Algorithm 919: A Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators. *ACM Trans. Math. Softw.*, 38(3):22:1–22:19, Apr. 2012. URL http://doi.acm.org/10.1145/2168773.2168781.

A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Number v. 44 in Applied Mathematical Sciences. Springer, 1992. URL http://books.google.de/books?id=sIAyOgM4R3kC.

D. A. Pope. An exponential method of numerical integration of ordinary differential equations. *Comm. ACM*, 6:491–493, 1963.

Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comp.*, 37(155):105–126, 1981. URL http://dx.doi.org/10.2307/2007504.

Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992. URL http://dx.doi.org/10.1137/0729014.

R. B. Sidje. EXPOKIT: a software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, Mar. 1998. URL http://doi.acm.org/10.1145/285861.285868.

M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *J. Comput. Phys.*, 213 (2):748–776, 2006. URL http://dx.doi.org/10.1016/j.jcp.2005.08.032.

M. Tokman. A new class of exponential propagation iterative methods of Runge-Kutta type (EPIRK). *J. Comput. Phys.*, 230(24):8762–8778, 2011. URL http://dx.doi.org/10.1016/j.jcp.2011.08.023.

M. Tokman, J. Loffeld, and P. Tranquilli. New adaptive exponential propagation iterative methods of Runge-Kutta type. *SIAM J. Sci. Comput.*, 34(5):A2650–A2669, 2012. URL http://dx.doi.org/10.1137/110849961.

L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer. Talbot quadratures and rational approximations. *BIT*, 46(3):653–670, 2006. URL http://dx.doi.org/10.1007/s10543-006-0077-9.