Computational Science Laboratory Technical Report CSLTR-1/2014
January 30, 2015

Paul Tranquilli and Adrian Sandu

"Exponential-Krylov methods for ordinary differential equations"

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: sandu@cs.vt.edu
Web: http://csl.cs.vt.edu

COMPUTATIONAL SCIENCE LABORATORY
Innovative Computational Solutions

VirginiaTech
Invent the Future

Exponential-Krylov methods for ordinary differential equations

Paul Tranquilli[a], Adrian Sandu[a]

[a]*Computational Science Laboratory, Department of Computer Science, Virginia Tech. Blacksburg, Virginia 24060*

Abstract

This paper develops a new family of exponential time discretization methods called exponential-Krylov (EXPK). The new schemes treat the time discretization and the Krylov based approximation of exponential matrix-vector products as a single computational process. The classical order conditions theory developed herein accounts for both the temporal and the Krylov approximation errors. Unlike traditional exponential schemes, EXPK methods require the construction of only a single Krylov space at each timestep. The number of basis vectors that guarantee the temporal order of accuracy does not depend on the application at hand. Numerical results show favorable properties of EXPK methods when compared to current exponential schemes.

1. Introduction

Many methods exist to numerically approximate the solution of initial value problems

$$\frac{dy}{dt} = f(t,y)\,, \quad t_0 \leq t \leq t_F\,, \quad y(t_0) = y_n\,; \quad y(t), f(t,y) \in \mathbb{R}^N\,. \tag{1}$$

Multistep methods make use of the solution at several previous timesteps to compute the solution at t_{n+1}, while Runge-Kutta methods interpolate the solution at several points between the current solution, t_n, and the future solution, t_{n+1}. In both cases implicit methods require the solution of (non-)linear system of equations at each time step. Much work has been done towards the acceleration of the solutions to these systems. Iterative Krylov-based linear algebra solvers are the typical choice for large-scale applications (1). The generalized minimal residual (GMRES) method [19] is the standard approach for constructing efficient solutions to linear systems arising throughout the integration of ODEs. Jacobian-free Newton-Krylov (JFNK) methods [8, 9] make use of a GMRES like solver within a Newton iteration to solve the nonlinear equations arising from Runge-Kutta and multistep methods.

Rosenbrock methods [4], a class of integrators coming from a linearization of Runge-Kutta methods, require only the solution of a linear system at each stage and are characterized by the explicit appearance of the Jacobian matrix

$$\mathbf{J}_n = \left.\frac{\partial f}{\partial y}\right|_{t=t_n, y=y_n}$$

in the method itself. Due to the approximate nature of solutions coming from iterative methods, the explicit appearance of $\mathbf{J}_n$ causes order reduction unless the system solution is very accurate. For this reason Rosenbrock-W methods [4, 14, 17], an extension of Rosenbrock methods allowing for arbitrary approximations of the matrix $\mathbf{J}_n$, have been developed.

Krylov-ROW methods [15, 18, 20, 26] couple Rosenbrock methods with Krylov based solvers for the linear systems arising therein. A multiple Arnoldi process is used to enrich the Krylov space at each stage, and the order of the underlying Rosenbrock method is preserved with modest requirements on the Krylov space size, independent of the dimension of the ODE system under consideration.

The authors have recently developed Rosenbrock-K methods [24] to pursue a similar goal. Krylov-ROW methods ensure the order results with standard Rosenbrock-W discretizations by adding requirements to the underlying Krylov space. In contradistinction, Rosenbrock-K methods guarantee the accuracy order through the use of a specific Krylov-based approximation of the Jacobian and the construction of new order conditions which take this approximation into account. Rosenbrock-K methods have substantially fewer order conditions than Rosenbrock-W methods allowing for the construction of schemes of higher order with fewer stages. More importantly, Rosenbrock-K methods give a strict lower bound on the number of Krylov basis vectors required for accuracy that depends only on the order of the method, and is completely independent of the dimension of the ODE system under consideration.

Exponential integrators [6, 22, 23] replace the need to construct solutions to a linear system, or equivalently approximate the rational matrix function times vector product $(\mathbf{I}_N - hA)^{-1} v$, with the similar, hopefully cheaper, requirement to approximate the exponential matrix times vector product $\exp(hA)v$. Like the solution of large linear systems, approximations of the matrix exponential times vectors are typically obtained using Krylov based methods.

In this paper we extend the ideas of the Rosenbrock-K methods presented in [24] to the particular set of exponential integrators discussed in Hochbruck, Lubich, and Selhofer [6] and introduce the new family of exponential-Krylov (exponential-K) methods. The new schemes require the construction of only a single Kyrlov basis at each timestep, as opposed to each stage in the case of standard exponential methods. Moreover, the required dimension of the subspace to guarantee the desired order of accuracy is independent of the system (1) under consideration.

The remainder of the paper is organized as follows. Section 2 presents the exponential-K framework and the Krylov approximation of the Jacobian used. Section 3 develops the order condition theory for the new exponential-K methods using both Butcher trees and B-series. Section 4 constructs a practical four stage, fourth order exponential-K method. Section 5 discusses alternative implementations of existing exponential methods, and Section 6 presents numerical results. Conclusions are drawn in Section 7.

2. Formulation of exponential-Krylov methods

2.1. Exponential-W integrators

The starting point of our investigation is following class of exponential-W integrators proposed in [6]

$$
\begin{aligned}
k_i &= \varphi(h\gamma\mathbf{A}_n)\left(h\,F_i + h\mathbf{A}_n\sum_{j=1}^{i-1}\gamma_{i,j}k_j\right), \\
F_i &= f\left(y_n + \sum_{j=1}^{i-1}\alpha_{i,j}k_j\right), \\
y_{n+1} &= y_n + \sum_{j=1}^{s}b_i k_i,
\end{aligned}
\tag{2}
$$

where $\mathbf{A}_n$ is either the matrix $\mathbf{J}_n$ or an approximation of it. Equation (2) formalizes explicit Runge-Kutta methods when $\varphi(z) = 1$, Rosenbrock methods when $\varphi(z) = 1/(1 - z)$, and exponential methods when $\varphi(z) = (e^z - 1)/z$. Note that similar to the Rosenbrock methods discussed before, equation (2) makes explicit use of the matrix $\mathbf{J}_n$, and so it is natural to explore conditions allowing for arbitrary approximations as in the case of Rosenbrock-W methods. A discussion of these methods and their order conditions is given in [6].

2.2. Exponential-K integrators

The new exponential-K methods proposed in this work have the same general form as exponential-W methods (2), but use a specific, Krylov based-approximation $\mathbf{A}_n$ of the Jacobian. To begin we construct

the M-dimensional Krylov space $\mathcal{K}_M$ where $M \ll N$ and

$$
\begin{aligned}
\mathcal{K}_M &= \operatorname{span}\left\{f_n, \mathbf{J}_n f_n, \mathbf{J}_n^2 f_n, \ldots, \mathbf{J}_n^{M-1} f_n\right\} \\
&= \operatorname{span}\left\{v_1, v_2, \ldots, v_M\right\}
\end{aligned}
\tag{3}
$$

using a modified Arnoldi iteration [25]. The Arnoldi iteration returns the matrix

$$
\mathbf{V} = [v_1, v_2, \ldots, v_M] \in \mathbb{R}^{N \times M}
$$

whose columns form an orthonormal basis of $\mathcal{K}_M$, and the upper Hessenberg matrix

$$
\mathbf{H} = \mathbf{V}^T \mathbf{J}_n \mathbf{V} \in \mathbb{R}^{M \times M}.
\tag{4}
$$

From these two matrices we construct the following Krylov-based approximation of the Jacobian

$$
\mathbf{A}_n = \mathbf{V} \mathbf{H} \mathbf{V}^T = \mathbf{V} \mathbf{V}^T \mathbf{J}_n \mathbf{V} \mathbf{V}^T.
\tag{5}
$$

The powers of this matrix have the following property.

Lemma 1 (Powers of $\mathbf{A}_n$). *For any $k = 0, 1, 2, \ldots$*

$$
\mathbf{A}_n^k = \mathbf{V} \mathbf{H}^k \mathbf{V}^T.
$$

PROOF. We give the proof of the Lemma by induction. As the base case we have that

$$
\mathbf{A}_n^1 = \mathbf{V} \mathbf{H} \mathbf{V}^T = \mathbf{V} \mathbf{H}^1 \mathbf{V}^T
$$

next we assume that $\mathbf{A}_n^{k-1} = \mathbf{V} \mathbf{H}^{k-1} \mathbf{V}^T$ and show that $\mathbf{A}_n^k = \mathbf{V} \mathbf{H}^k \mathbf{V}^T$.

$$
\mathbf{A}_n^k = \mathbf{A}_n \mathbf{A}_n^{k-1} = \mathbf{V} \mathbf{H} \mathbf{V}^T \left(\mathbf{V} \mathbf{H}^{k-1} \mathbf{V}^T\right) = \mathbf{V} \mathbf{H} \left(\mathbf{V}^T \mathbf{V}\right) \mathbf{H}^{k-1} \mathbf{V}^T
$$

because V is an orthonormal matrix $\mathbf{V}^T \mathbf{V} = \mathbf{I}_N$, and so

$$
\mathbf{A}_n^k = \mathbf{V} \mathbf{H}^k \mathbf{V}^T
$$

$\square$

The construction of exponential integrators uses matrix functions of the form $\varphi_k(h\gamma \mathbf{A}_n)$, where the functions are defined by

$$
\varphi_k(z) = \int_0^1 e^{z(1-\theta)} \frac{\theta^{k-1}}{(k-1)!} \, d\theta, \quad k = 0, 1, 2, \ldots
$$

and satisfy the recurrence relation

$$
\varphi_{k+1}(z) = \frac{\varphi_k(z) - 1/k!}{z}, \quad \varphi_k(0) = \frac{1}{k!}.
$$

The matrix functions have the following property.

Lemma 2 (Matrix functions of the approximate Jacobian).

$$
\varphi_k(h\gamma \mathbf{A}_n) = \frac{1}{k!}(\mathbf{I}_N - \mathbf{V} \mathbf{V}^T) + \mathbf{V} \varphi_k(h\gamma \mathbf{H}) \mathbf{V}^T \quad k = 1, 2, \ldots
$$

3

PROOF. It is possible to expand $\varphi_k(z)$ as a Taylor series [11]:

$$\varphi_k(z) = \sum_{i=0}^{\infty} c_i \frac{z^i}{(k+i)!} \tag{6}$$

We have that

$$\varphi_k(h\gamma\mathbf{A}_n) = \frac{1}{k!}\mathbf{I}_N + \sum_{i=1}^{\infty} \frac{(h\gamma)^i}{(k+i)!}\mathbf{A}_n^i$$

after applying Lemma 1 we obtain

$$\varphi_k(h\gamma\mathbf{A}_n) = \frac{1}{k!}\mathbf{I}_N + \sum_{i=1}^{\infty} \frac{(h\gamma)^i}{(k+i)!}\mathbf{V}\mathbf{H}^i\mathbf{V}^T.$$

Similarly we can expand $\mathbf{V}\varphi_k(h\gamma H)\mathbf{V}^T$ as

$$\mathbf{V}\varphi_k(h\gamma\mathbf{H})\mathbf{V}^T = \frac{1}{k!}\mathbf{V}\mathbf{V}^T + \sum_{i=1}^{\infty} \frac{(h\gamma)^i}{(k+i)!}\mathbf{V}\mathbf{H}^i\mathbf{V}^T,$$

taking the difference we see that

$$\varphi_k(h\gamma\mathbf{A}_n) - \mathbf{V}\varphi_k(h\gamma\mathbf{H})\mathbf{V}^T = \frac{1}{k!}\left(\mathbf{I}_N - \mathbf{V}\mathbf{V}^T\right).$$

Finally we move $\mathbf{V}\varphi_k(h\gamma H)\mathbf{V}^T$ across the equality to obtain

$$\varphi_k(h\gamma\mathbf{A}_n) = \frac{1}{k!}\left(\mathbf{I}_N - \mathbf{V}\mathbf{V}^T\right) + \mathbf{V}\varphi_k(h\gamma\mathbf{H})\mathbf{V}^T \tag{7}$$

$\square$

To finish the derivation of a reduced form for the exponential-Krylov integrator (2) we introduce the following notation

$$k_i = \underbrace{\mathbf{V}\lambda_i}_{\in\mathcal{K}_M} + \underbrace{\mu_i}_{\in\mathcal{K}_M^\perp}, \quad F_i = \underbrace{\mathbf{V}\psi_i}_{\in\mathcal{K}_M} + \underbrace{\delta_i}_{\in\mathcal{K}_M^\perp}$$

where $\mathbf{V}\lambda_i$ and $\mathbf{V}\psi_i$ represent the components of k_i and F_i which reside in the Krylov subspace $\mathcal{K}_M$, and similarly μ_i and δ_i are the components residing in the space orthogonal to $\mathcal{K}_M$. Insert equation (7) and the split forms of k_i and F_i into the general method formulation (2) to obtain

$$\mathbf{V}\lambda_i + \mu_i = \mathbf{V}\left(\varphi(h\gamma\mathbf{H})\psi_i + h\mathbf{H}\sum_{j=1}^{i-1}\gamma_{i,j}\lambda_j\right) + \delta_i.$$

This leads to the following equation for the reduced stage vector

$$\lambda_i = \varphi(h\gamma\mathbf{H})\left(\psi_i + h\mathbf{H}\sum_{j=1}^{i-1}\gamma_{i,j}\lambda_j\right). \tag{8}$$

The full stage values can be recovered as

$$k_i = \mathbf{V}\lambda_i + (F_i - \mathbf{V}\psi_i). \tag{9}$$

A single step of an autonomous exponential-K method is given in Algorithm 1.

4

Algorithm 1 One step of an autonomous exponential-K integrator

1: Compute $\mathbf{H}$ and $\mathbf{V}$ using the N-dimensional Arnoldi process [25]
2: **for** $i = 1, \ldots, s$ **do**   ▷ For each stage, in succession

$$
\begin{aligned}
F_i &= f\left(y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j\right) \\
\psi_i &= \mathbf{V}^T F_i \\
\lambda_i &= \varphi\left(h\gamma\mathbf{H}\right)\left(h\psi_i + h\mathbf{H}\sum_{j=1}^{i-1}\gamma_{i,j}\lambda_j\right) \\
k_i &= \mathbf{V}\lambda_i + h\left(F_i - \mathbf{V}\psi_i\right)
\end{aligned}
$$

3: **end for**
4: $y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i$

Remark 1. *Because the matrix $\mathbf{H}$ has dimension $M \times M$ direct methods can be used to compute the matrix function $\varphi(h\gamma\mathbf{H})$. In the case of Rosenbrock methods where $\varphi(z) = 1/(1-z)$ a direct LU-decomposition can be used. For exponential methods where $\varphi(z) = (e^z - 1)/z$ a Pade approximation [5] can be utilized. Furthermore, a single matrix function needs to be evaluated at each step when the matrices $\mathbf{H}$ and $\mathbf{V}$ are constructed.*

Remark 2. *We have only given here the autonomous form of an exponential-K method. A non-autonomous form is possible by constructing an extended ODE system and the corresponding Jacobian. This construction follows closely the treatment given in [24], where an $N+1$ dimensional Arnoldi iteration is discussed.*

3. Order conditions for exponential-K methods

We construct classical order conditions for exponential-K methods. To this end we match the Taylor series expansion of the numerical and exact solutions up to a specified order. Butcher-trees[3] are an established method of representing terms in the Taylor series expansions of Runge-Kutta like methods. The derivation of order conditions for K-methods is an extension of the framework developed for W-methods. The theory for W methods is constructed using TW-trees, a subclass of P-trees, which are themselves an extension of T-trees that allow for two different colored nodes.

$$
TW = \left\{ \begin{array}{ll} P\text{-trees:} & \text{end vertices are meagre, and} \\ & \text{fat vertices are singly branched} \end{array} \right\}
$$

In the context of TW (and TK)-trees a meagre, or solid, node represents an appearance of the exact Jacobian matrix $\mathbf{J}_n$, while a fat, or empty, node represents the appearance of the approximate Jacobian matrix $\mathbf{A}_n$. Each tree represents a single elementary differential in the Taylor series of either the exact or numerical solutions of the ODE.

A fundamental component of our derivation of order conditions for exponential-K methods are B-series, a way of representing an expansion in trees, or elementary differentials, as a sequence of real numbers. A mapping $\mathsf{a} : TW \cup \{\emptyset\} \to \mathbb{R}$ represents the series

$$
B(\mathsf{a}, y) = \mathsf{a}(\emptyset)y + \sum_{\tau \in TW} a(\tau)\frac{h^{|\tau|}}{\sigma(\tau)}F(\tau)(y).
$$

Here τ are TW-trees; the order $|\tau|$, and the symmetry $\sigma(\tau)$ of a tree are defined in the same way as for single colored trees [3, 23], and $F(\tau)$ is the elementary differential belonging to tree τ as in figure 1. Similarly we

τ	 ● j	● k ● j	● k ○ j	l ● ● k ● j	l ● ● k ● j
$F(\tau)$	f^J	$f_K^J f^K$	$\mathbf{A}_{JK} f^K$	$f_{KL}^J f^K f^L$	$f_K^J f_L^K f^L$
$\mathsf{a}(\tau)$	x_1	x_2	x_3	x_4	x_5
$B^{\#}\left(hf(B(\mathsf{a},y))\right)$	1	x_1	0	x_1^2	x_2
$B^{\#}\left(h\mathbf{A}B(\mathsf{a},y)\right)$	0	0	x_1	0	0
$B^{\#}\left(\varphi(h\gamma\mathbf{A})B(\mathsf{a},y)\right)$	x_1	x_2	$x_3 + c_1 x_1$	x_4	x_5

τ	l ● ○ k ● j	l ● ● k ○ j	l ● ○ k ○ j	k ● ● l ● m ● j	● m k ● ● l ● j
$F(\tau)$	$f_K^J \mathbf{A}_{KL} f^L$	$\mathbf{A}_{JK} f_L^K f^L$	$\mathbf{A}_{JK} \mathbf{A}_{KL} f^L$	$f_{KLM}^J f^K f^L f^M$	$f_{KL}^J f_M^L f^M f^K$
$\mathsf{a}(\tau)$	x_6	x_7	x_8	x_9	x_{10}
$B^{\#}\left(hf(B(\mathsf{a},y))\right)$	x_3	0	0	x_1^3	$x_1 x_2$
$B^{\#}\left(h\mathbf{A}B(\mathsf{a},y)\right)$	0	x_2	x_3	0	0
$B^{\#}\left(\varphi(h\gamma\mathbf{A})B(\mathsf{a},y)\right)$	x_6	$x_7 + c_1 x_2$	$x_8 + c_1 x_3 + c_2 x_1$	x_9	x_{10}

τ	● m k ● ○ l ● j	m ● ● l ● k ● j	m ● ● l ● k ○ j	● m l ● ● k ● j	● m l ○ ● k ● j
$F(\tau)$	$f_{KL}^J \mathbf{A}_{LM} f^M f^K$	$f_K^J f_{LM}^K f^M f^L$	$\mathbf{A}_{JK} f_{LM}^K f^L f^M$	$f_K^J f_L^K f_M^L f^M$	$f_K^J f_L^K \mathbf{A}_{LM} f^M$
$\mathsf{a}(\tau)$	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
$hf(B(\mathsf{a},y))$	$x_1 x_3$	x_4	0	x_5	x_6
$B^{\#}\left(h\mathbf{A}B(\mathsf{a},y)\right)$	0	0	x_4	0	0
$B^{\#}\left(\varphi(h\gamma\mathbf{A})B(\mathsf{a},y)\right)$	x_{11}	x_{12}	$x_{13} + c_1 x_4$	x_{14}	x_{15}

Figure 1: TW-trees up to order four (part one of two).

introduce the operator $B^{\#}(f)$, which takes as input a function (which can be represented by a series) and returns the B-series coefficients

$$B^{\#}\left(B(\mathsf{a},y)\right) = \mathsf{a}.$$

Figure 1 shows all TW-trees to order four, the coefficients of a generic B-series $B(\mathsf{a},y)$, the result of composing $B(\mathsf{a},y)$ with the function $f(y)$, the result of a multiplication of $B(\mathsf{a},y)$ with the Krylov approximation matrix $\mathbf{A}_n$, and the result of a multiplication of $B(\mathsf{a},y)$ by $\varphi(h\gamma\mathbf{A}_n)$. The full details of the composition of B-series can be found in [2], while details of products with the Jacobian and φ-functions can be found in [1].

Because K-methods are an extension of W-methods we first construct order conditions for the W-methods, then prove two lemmas that allow us to obtain the specific exponential-K order conditions. We follow a similar derivation procedure to that outlined in [16]. Throughout the derivation we track the progress of several truncated B-series which include all terms up to order four. These series have 21 terms corresponding to the TW-trees shown in Figure 1.

We begin the construction of order conditions for the W-method with a truncated B-series for y_n

$$B^{\#}\left(y(t_0)\right) = \mathsf{a}_0 = \left\{\mathsf{a}_0(\emptyset) = 1, x_i = 0 \ \ \forall i = 1, \ldots, 21\right\},$$

6

τ			
$F(\tau)$	$f_K^J \mathbf{A}_{KL} f_M^L f^M$	$f_K^J \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$	$\mathbf{A}_{JK} f_L^K f_M^L f^M$
$\mathsf{a}(\tau)$	x_{16}	x_{17}	x_{18}
$B^\#\left(hf(B(\mathsf{a},y))\right)$	x_7	x_8	0
$B^\#\left(h\mathbf{A}B(\mathsf{a},y)\right)$	0	0	x_5
$B^\#\left(\varphi(h\gamma\mathbf{A})B(\mathsf{a},y)\right)$	x_{16}	x_{17}	$x_{18}+c_1 x_5$
τ			
$F(\tau)$	$\mathbf{A}_{JK} f_L^K \mathbf{A}_{LM} f^M$	$\mathbf{A}_{JK} \mathbf{A}_{KL} f_M^L f^M$	$\mathbf{A}_{JK} \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$
$\mathsf{a}(\tau)$	x_{19}	x_{20}	x_{21}
$B^\#\left(hf(B(\mathsf{a},y))\right)$	0	0	0
$B^\#\left(h\mathbf{A}B(\mathsf{a},y)\right)$	x_6	x_7	x_8
$B^\#\left(\varphi(h\gamma\mathbf{A})B(\mathsf{a},y)\right)$	$x_{19}+c_1 x_6$	$x_{20}+c_1 x_7+c_2 x_2$	$x_{21}+c_1 x_8+c_2 x_3+c_3 x_1$

Figure 2: TW-trees up to order four (part two of two).

and then progress through individual stages of the W-method in equation (2), making use of the formulas from Figure 1 to construct the resultant B-series of the composition and multiplication operations. Algorithm 2 gives a method of constructing the B-series of the numerical solution y_n that approximates the exact solution $y(t_0 + h)$. Note that the sum of two B-series is another B-series with coefficients equal to the sum of individual coefficients of the series being combined. Similarly, the product of a B-series with a scalar is a new series with each coefficient multiplied by the scalar.

The order conditions of the W-methods are obtained by matching the B-series coefficients of the exact solution $B^\#\left(y(t_n + h)\right)$ with those of the numerical solution $B^\#\left(\mathsf{y}_{n+1}\right)$ up to a specified order. Keeping in mind that we do not ultimately seek order conditions for a W-method itself, that they are simply a means to an end, we look now at the process for obtaining order conditions of the K-method from this result.

The extension of the theory of TW-trees to TK-trees is done in [24]. This extension allows us to "recolor" all linear sub-trees (possessing only singly branched nodes) of the TW-trees and to substantially reduce the number of required conditions. This is done using Lemmas 3 and 4, taken from [24], and repeated here without proof.

Lemma 3 (Property of the Krylov approximate Jacobian (5) [24]). *For any $0 \le k \le M-1$ it holds that*

$$\mathbf{A}_n^k f_n = \mathbf{J}_n^k f_n\,,$$

where $M = dim(\mathcal{K}_M)$.

Lemma 4 (Property of elementary differentials using the approximation (5) [24]). *When the Krylov approximation matrix (5) is used in equation (2), all linear TW-trees of order $k \le M$ correspond to a single elementary differential, regardless of the color of their nodes.*

TK-trees are the result of an application of Lemmas 3 and 4 to reduce the set of TW-trees that need to be considered in the order conditions when the Krylov approximation matrix (5) is used [24].

Algorithm 2 Construction of B-series of the numerical solution of a fourth-order, s stage exponential-W-method (2)

> **for** $i = 1, \ldots, s$ **do**
> $\quad$ u $= $ a$_0$
> $\quad$ **for** $j = 1, \ldots, i - 1$ **do**
> $\quad\quad$ u $= $ u $+ \alpha_{i,j} \cdot$ k$_i$
> $\quad$ **end for**
> $\quad$ q $= B^{\#} \left(h f \left(B(\mathsf{u}, y) \right) \right)$
> $\quad$ **for** $j = 1, \ldots, i - 1$ **do**
> $\quad\quad$ q $= $ q $+ \gamma_{i,j} \cdot B^{\#} \left(h \mathbf{A}_n B(\mathsf{k}_i, y) \right)$
> $\quad$ **end for**
> $\quad$ k$_i = B^{\#} \left(\varphi(h\gamma \mathbf{A}_n) B(\mathsf{q}, y) \right)$
> **end for**
> a$_n = $ a$_0$
> **for** $i = 1, \ldots, s$ **do**
> $\quad$ a$_n = $ a$_n + b_i \cdot$ k$_i$
> **end for**

Definition 1 (TK-trees [24]).

$$
\begin{aligned}
TK \quad &= \quad \{TW\text{-trees: no linear sub-tree has a fat root}\} \\
TK(k) \quad &= \quad \{TW\text{-trees: no linear sub-tree of order} \\
&\qquad \text{smaller than or equal to } k \text{ has a fat root}\} .
\end{aligned}
$$

Figure 3 shows all TK-trees to order four and the corresponding exponential-K order conditions. Note that there are only nine TK-trees as opposed to the original twenty TW-trees. There is only one additional order condition compared to methods which make use of the exact Jacobian, and this condition corresponds to a tree which has a doubly-branched node occurring as a descendant of a fat node.

Remark 3. *The order conditions given here are only for the exponential-K methods, but the same process can be used to rederive the Rosenbrock-K conditions given in [24] through the use of different c_n in the Taylor expansion of $\varphi(h\gamma \mathbf{A}_n)$ in equation (6).*

Theorem 1 (Order conditions for exponential-K methods). *An exponential-K method has order p iff the underlying Krylov space (3) has dimension $M \geq p$, and the following order conditions hold:*
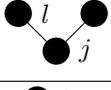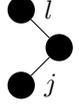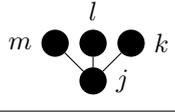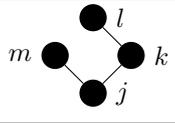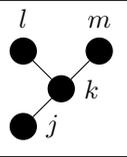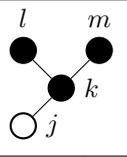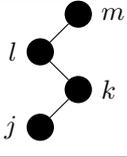
$$
\sum_j b_j \, \Phi_j(\tau) = P_\tau(\tau) \quad \forall \, \tau \in TK \quad with \ |\tau| \leq p \,. \tag{10}
$$

Here $|\tau|$ is the order, or number of vertices of the tree τ, and $\Phi_j(\tau)$ and $P_\tau(\tau)$ are computed using Algorithm 2; they are shown in Figure 3 for $p \leq 4$.

PROOF. The proof follows from our discussion, the near equivalence of order conditions for exponential-W and Rosenbrock-W methods [6], and from the order conditions of Rosenbrock-W methods [4, Theorem 7.7]. $\qquad \square$

Remark 4 (Stiff order conditions.). *This section has developed classical order conditions that explain the accuracy of the methods on non-stiff problems. The behavior of the methods when applied to very stiff problems may be different, e.g., true to order reduction. A stiff order conditions theory for exponential methods has been proposed by Luan and Ostermann [13]. The development of stiff order conditions for exponential-K methods falls outside the scope of this paper.*

Figure 3: TK-trees and exponential-K conditions up to order four.

τ	$F(\tau)$	$\Phi(\tau)$	$P_\tau(\gamma)$
$\bullet\, j$	f^J	1	1
(tree)	$f_K^J f^K$	$\sum \beta_{j,k}$	$1/2\,(1-\gamma)$
(tree)	$f_{KL}^J f^K f^L$	$\sum \alpha_{j,k}\alpha_{j,l}$	$1/3$
(tree)	$f_K^J f_L^K f^L$	$\sum \beta_{j,k}\beta_{k,l}$	$1/3(1/2-\gamma)(1-\gamma)$
(tree)	$f_{KLM}^J f^K f^L f^M$	$\sum \alpha_{j,k}\alpha_{j,l}\alpha_{jm}$	$1/4$
(tree)	$f_{KM}^J f_L^K f^L f^M$	$\sum \alpha_{j,k}\beta_{k,l}\alpha_{j,m}$	$1/8 - \gamma/6$
(tree)	$f_K^J f_{LM}^K f^L f^M$	$\sum \alpha_{j,k}\alpha_{k,m}\alpha_{k,l}$	$1/12$
(tree)	$\mathbf{A}_{JK} f_{LM}^K f^L f^M$	$\sum \gamma_{j,k}\alpha_{k,m}\alpha_{k,l}$	$-\gamma/6$
(tree)	$f_K^J f_L^K f_M^L f^M$	$\sum \beta_{j,k}\beta_{k,l}\beta_{l,m}$	$1/4(1/3-\gamma)(1/2-\gamma)(1-\gamma)$

Remark 5 (Stability considerations.). *The numerical stability of exponential-K solutions depends on the choice of Krylov space. Intuitively, the size of the K space should be large enough to cover the stiff subspace of the system.Note that traditional exponential methods focus on accurate computations of matrix-exp-vector products (e.g., by monitoring residuals), but do not account explicitly for the impact of Krylov approximations on stability. The large number of basis vectors required to achieve accurate matrix function vector products favors stability. In our case a small K dimension ensures accuracy, so the stability needs to be considered separately. An automatic procedure to select dimension such as to achieve stability is important, but falls outside the scope here.*

4. An exponential-K method of order four

We now construct an exponential-K method of order four. As before we consider the case where $\gamma_{i,i} = \gamma$ for all stages i and denote

$$\beta_{i,j} = \alpha_{i,j} + \gamma_{i,j}, \quad \beta_i' = \sum_{j=1}^{i-1} \beta_{i,j}.$$

The following nine non-linear equations arise from the order conditions of a four stage, fourth order exponential-K method

$$
\begin{array}{rlll}
(a) & b_1 + b_2 + b_3 + b_4 & = 1 & \\
(b) & b_2\beta_2' + b_3\beta_3' + b_4\beta_4' & = \frac{1}{2}(1-\gamma) & = p_{21}(\gamma) \\
(c) & b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 & = \frac{1}{3} & \\
(d) & b_3(\beta_{3,2}\beta_2') + b_4(\beta_{4,2}\beta_2' + \beta_{4,3}\beta_3') & = \frac{1}{3}(\frac{1}{2}-\gamma)(1-\gamma) & = p_{3,2}(\gamma) \\
(e) & b_2\alpha_2^3 + b_3\alpha_3^3 + b_4\alpha_4^3 & = \frac{1}{4} & \\
(f) & b_3\alpha_{3,2}\beta_2' + b_4(\alpha_{4,2}\beta_2' + \alpha_{4,3}\beta_3) & = \frac{1}{8} - \frac{1}{6}\gamma & = p_{4,2}(\gamma) \\
(g_1) & b_3\alpha_{3,2}\alpha_2^2 + b_4(\alpha_{4,2}\alpha_2^2 + \alpha_{4,3}\alpha_3^2) & = \frac{1}{12} & \\
(g_2) & b_3\gamma_{3,2}\alpha_2^2 + b_4(\gamma_{4,2}\alpha_2^2 + \gamma_{4,3}\alpha_3^2) & = -\frac{1}{6}\gamma & \\
(h) & b_4\beta_{4,3}\beta_{3,2}\beta_2' & = \frac{1}{4}(\frac{1}{3}-\gamma)(\frac{1}{2}-\gamma)(1-\gamma) & = p_{4,4}(\gamma)
\end{array}
\tag{11}
$$

If we now set

$$p_{4,3}(\gamma) = \frac{1}{12} - \frac{1}{6}\gamma,$$

we can follow exactly the solution procedure given in [24] for obtaining the ROK4a method, where we make use of the $p_{i,j}$ given above, and as suggested in [6] to guarantee exact solutions for linear ODEs choose γ as the reciprocal of an integer. For the EXPK method given in Table 1 we make the arbitrary choices

$$\gamma = \frac{1}{4}, \quad b_3 = 0, \quad \alpha_2 = 1, \quad \alpha_3 = \alpha_4 = \frac{1}{2}, \quad \beta_{4,3} = -\frac{1}{4}.$$

5. Alternative implementations of existing exponential methods

We now consider alternative implementations of previously derived methods EXP4 [6] and EROW4 [7]. These reformulations make use of only a single Krylov subspace projection per time step and exploit the B-series analysis of Section 3.

$\gamma = \frac{1}{4}$					
$\alpha_{2,1}$	$=$	1	$\gamma_{2,1}$	$=$	$\frac{7}{8}$
$\alpha_{3,1}$	$=$	$\frac{41}{80}$	$\gamma_{3,1}$	$=$	$\frac{1}{16}$
$\alpha_{3,2}$	$=$	$\frac{1}{80}$	$\gamma_{3,2}$	$=$	0
$\alpha_{4,1}$	$=$	$\frac{1}{4}$	$\gamma_{4,1}$	$=$	$-\frac{1}{32}$
$\alpha_{4,2}$	$=$	$\frac{1}{12}$	$\gamma_{4,2}$	$=$	$\frac{1}{24}$
$\alpha_{4,3}$	$=$	$\frac{1}{6}$	$\gamma_{4,3}$	$=$	$-\frac{5}{12}$
b_1	$=$	$\frac{1}{6}$	$\widehat{b}_1$	$=$	$\frac{8}{3}$
b_2	$=$	$\frac{1}{6}$	$\widehat{b}_2$	$=$	1
b_3	$=$	0	$\widehat{b}_3$	$=$	$-\frac{8}{3}$
b_4	$=$	$\frac{2}{3}$	$\widehat{b}_4$	$=$	0

Table 1: Coefficients of EXPK, a fourth order exponential-K method.

The method EXP4 [6] has the alternative formulation:

$$k_1 = \varphi_1\left(\frac{1}{3}h\mathbf{A}_n\right)f(y_n), \quad k_2 = \varphi_1(\frac{2}{3}h\mathbf{A}_n)f(y_n), \quad k_3 = \varphi_1(h\mathbf{A}_n)f(y_n), \tag{12a}$$

$$w_4 = \frac{-7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3, \tag{12b}$$

$$u_4 = y_n + hw_4, \quad d_4 = f(u_4) - f(y_n) - h\mathbf{A}_n w_4, \tag{12c}$$

$$k_4 = \varphi_1\left(\frac{1}{3}h\mathbf{A}_n\right)d_4, \quad k_5 = \varphi_1(\frac{2}{3}h\mathbf{A}_n)d_4, \quad k_6 = \varphi_1(h\mathbf{A}_n)d_4, \tag{12d}$$

$$w_7 = \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6), \tag{12e}$$

$$u_7 = y_n + hw_7, \quad d_7 = f(u_7) - f(y_n) - h\mathbf{A}_n w_7, \tag{12f}$$

$$k_7 = \varphi_1\left(\frac{1}{3}h\mathbf{A}_n\right)d_7, \tag{12g}$$

$$y_1 = y_n + h\left(k_3 + k_4 - \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7\right). \tag{12h}$$

The method EROW4 [7] has the alternative formulation

$$k_1 = \varphi_1(\frac{1}{2}h\mathbf{A}_n)f(y_n), \tag{13a}$$

$$w_2 = \frac{1}{2}k_1, \tag{13b}$$

$$u_2 = y_n + hw_2, \quad d_2 = f(u_2) - f(y_n) - h\mathbf{A}_n w_2, \tag{13c}$$

$$k_2 = \varphi_1(h\mathbf{A}_n)f(y_n), \quad k_3 = \varphi_1(h\mathbf{A}_n)d_2, \tag{13d}$$

$$w_4 = k_2 + k_3, \tag{13e}$$

$$u_4 = y_n + hw_4, \quad d_4 = f(u_4) - f(y_n) - h\mathbf{A}_n w_4, \tag{13f}$$

$$k_4 = \varphi_3(h\mathbf{A}_n)d_2, \quad k_5 = \varphi_4(h\mathbf{A}_n)d_2, \quad k_6 = \varphi_3(h\mathbf{A}_n)d_4, \quad k_7 = \varphi_4(h\mathbf{A}_n)d_4, \tag{13g}$$

$$y_{n+1} = y_n + h(k_2 + 16k_4 - 48k_5 - 2k_6 + 12k_7) \tag{13h}$$

We implement these methods in three different forms: first, in the standard way outlined in the literature [6, 22, 23]; second, entirely in the reduced space such as given in (12), (13), and in [24]; and finally, using only a single Krylov projection to approximate the φ functions. These implementations are discussed below.

5.1. Standard implementation

The primary feature of a standard implementation of an exponential method is the approximation of φ functions using Krylov subspaces. For a term of the form $\varphi(h\mathbf{A}_n)b$ this is done by projecting $\varphi(h\mathbf{A}_n)$ and b onto the space $\mathcal{K}_M(\mathbf{A}_n, b) = \text{span}\left\{b, \mathbf{A}_n b, \mathbf{A}_n^2 b, \ldots, \mathbf{A}_n^{M-1} b\right\}$ as follows

$$\varphi(h\mathbf{A}_n)b \approx \mathbf{V}\mathbf{V}^T \varphi(h\mathbf{A}_n)\mathbf{V}\mathbf{V}^T b. \tag{14}$$

Note that $V^T b = \|b\|_2 e_1$, where e_1 is the first canonical basis vector. Making use of equation (4) we obtain the final Krylov subspace approximation

$$\varphi(h\mathbf{A}_n)b \approx \|b\|_2 V\varphi(h\mathbf{H})e_1 . \tag{15}$$

This approximation is computed as in [21], in which the exponential of an augmented matrix $\widetilde{\mathbf{H}}$ is constructed [5] and (15) is read off from this result.

Remark 6. *The standard implementation requires the construction of a new Krylov space for each vector b operated on by a φ function, as well as the evaluation of a small matrix exponential to compute each $\varphi(h\gamma\mathbf{H})$ function. Both* EXP4 *and* EROW4 *require the construction of three Krylov subspaces and the evaluation of seven small matrix exponentials.*

5.2. K-type implementation

K-type implementations of EXP4 and EROW4 follow the style of [24] and Section 1. For each $k_i \in \mathbb{R}^N$ we create a corresponding $\lambda_i = \mathbf{V}^T k_i \in \mathbb{R}^M$, similarly $\sigma_i = \mathbf{V}^T w_i \in \mathbb{R}^M$, and evaluate all linear algebra operations, including Jacobian-vector products, in the reduced space. Further, we construct only a single Krylov subspace and perform full matrix computations of three φ function evaluations in the case of EXP4, and four in the case of EROW4.

The K-type implementation of EXP4, called EXP4K, is:

$$\psi_0 = \mathbf{V}^T f(y_n), \quad f_0^\perp = f - \mathbf{V}\psi_0,$$

$$\lambda_1 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\psi_0, \quad \lambda_2 = \varphi_1(\tfrac{2}{3}h\mathbf{H})\psi_0, \quad \lambda_3 = \varphi_1(h\mathbf{H})\psi_0,$$

$$k_1 = \mathbf{V}\lambda_1 + f_0^\perp, \quad k_2 = \mathbf{V}\lambda_2 + f_0^\perp, \quad k_3 = \mathbf{V}\lambda_3 + f_0^\perp, \tag{16a}$$

$$w_4 = \frac{-7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3, \quad \sigma_4 = \frac{-7}{300}\lambda_1 + \frac{97}{150}\lambda_2 - \frac{37}{300}\lambda_3, \tag{16b}$$

$$u_4 = y_n + hw_4, \quad \psi_4 = \mathbf{V}^T f(u_4), \quad f_4^\perp = f(u_4) - \mathbf{V}\psi_4, \quad \delta_4 = \psi_4 - \psi_0 - h\mathbf{H}\sigma_4, \tag{16c}$$

$$\lambda_4 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\delta_4, \quad \lambda_4 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\delta_4, \quad \lambda_4 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\delta_4,$$

$$k_4 = \mathbf{V}\lambda_4 + f_4^\perp - f_0^\perp, \quad k_5 = \mathbf{V}\lambda_5 + f_4^\perp - f_0^\perp, \quad k_6 = \mathbf{V}\lambda_6 + f_4^\perp - f_0^\perp, \tag{16d}$$

$$w_7 = \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6),$$

$$\sigma_7 = \frac{59}{300}\lambda_1 - \frac{7}{75}\lambda_2 + \frac{269}{300}\lambda_3 + \frac{2}{3}(\lambda_4 + \lambda_5 + \lambda_6), \tag{16e}$$

$$u_7 = y_n + hw_7, \quad \psi_7 = \mathbf{V}^T f(u_7), \quad f_7^\perp = f(u_7) - V\psi_7, \quad \delta_7 = \psi_7 - \psi_0 - h\mathbf{H}\sigma_7, \tag{16f}$$

$$\lambda_7 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\delta_7, \quad k_7 = \mathbf{V}^T\lambda_7 + f_7^\perp - f_0^\perp, \tag{16g}$$

$$y_1 = y_n + h\left(k_3 + k_4 - \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7\right). \tag{16h}$$

12

The K-type implementation of EROW4, called EROW4K, is:

$$\psi_0 = \mathbf{V}^T f(y_n), \quad f_0^\perp = f(y_n) - \mathbf{V}\psi_0,$$

$$\lambda_1 = \varphi_1(\frac{1}{2}h\mathbf{H})\psi_0, \quad k_1 = \mathbf{V}\lambda_1 + f_0^\perp, \tag{17a}$$

$$w_2 = \frac{1}{2}k_1, \quad \sigma_2 = \frac{1}{2}\lambda_1, \tag{17b}$$

$$u_2 = y_n + hw_2, \quad \psi_2 = \mathbf{V}^T f(u_2), \quad f_2^\perp = f(u_2) - \mathbf{V}\psi_2, \quad \delta_2 = \psi_2 - \psi_0 - h\mathbf{H}\sigma_2, \tag{17c}$$
$$\lambda_2 = \varphi_1(h\mathbf{H})\psi_0, \quad k_2 = \mathbf{V}\lambda_2 + f_0^\perp,$$

$$\lambda_3 = \varphi_1(h\mathbf{H})\delta_2, \quad k_3 = \mathbf{V}\lambda_3 + f_2^\perp - f_0^\perp, \tag{17d}$$

$$w_4 = k_2 + k_3, \quad \sigma_4 = \lambda_2 + \lambda_3, \tag{17e}$$

$$u_4 = y_n + hw_4, \quad \psi_4 = \mathbf{V}^T f(u_4), \quad f_4^\perp = f(u_4) - V\psi_4, \quad \delta_4 = \psi_4 - \psi_0 - h\mathbf{H}\sigma_4, \tag{17f}$$

$$\lambda_4 = \varphi_3(h\mathbf{H})\delta_2, \quad k_4 = \mathbf{V}\lambda_4 + \frac{1}{3!}\left(f_2^\perp - f_0^\perp\right),$$

$$\lambda_5 = \varphi_4(h\mathbf{H})\delta_2,, \quad k_5 = \mathbf{V}\lambda_5 + \frac{1}{4!}\left(f_2^\perp - f_0^\perp\right),$$

$$\lambda_6 = \varphi_3(h\mathbf{H})\delta_4, \quad k_6 = \mathbf{V}\lambda_6 + \frac{1}{3!}\left(f_4^\perp - f_0^\perp\right),$$

$$\lambda_7 = \varphi_4(h\mathbf{H})\delta_4, \quad k_7 = \mathbf{V}\lambda_7 + \frac{1}{4!}\left(f_4^\perp - f_0^\perp\right), \tag{17g}$$

$$y_{n+1} = y_n + h\left(k_2 + 16k_4 - 48k_5 - 2k_6 + 12k_7\right) \tag{17h}$$

5.3. Single projection implementation

The results of Section 3 imply that a single Krylov subspace need to be computed per time step guarantee the order of accuracy. In contradistinction the standard implementation constructs several Krylov spaces, primarily due to the use of residuals indicating how accurately the matrix function times vector products have been approximated. In the single projection implementation we construct only the one subspace, and similarly to K-type implementation compute the full matrix result of the φ functions. The implementation differs from the K-type implementation in that the linear algebra operations, including Jacobian-vector products, are computed in the full space.

The single projection implementation of EXP4, called EXP4SP, is:

$$\psi_0 = \mathbf{V}^T f(y_n), \quad f_0^\perp = f(y_n) - \mathbf{V}\psi_0,$$

$$\lambda_1 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\psi_0, \quad \lambda_2 = \varphi_1(\tfrac{2}{3}h\mathbf{H})\psi_0, \quad \lambda_3 = \varphi_1(h\mathbf{H})\psi_0,$$

$$k_1 = \mathbf{V}\lambda_1 + f_0^\perp, \quad k_2 = \mathbf{V}\lambda_2 + f_0^\perp, \quad k_3 = \mathbf{V}\lambda_3 + f_0^\perp, \tag{18a}$$

$$w_4 = \frac{-7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3, \tag{18b}$$

$$u_4 = y_n + hw_4, \quad d_4 = f(u_4) - f(y_n) - h\mathbf{J}_n w_4, \tag{18c}$$

$$\psi_4 = \mathbf{V}^T d_4, \quad d_4^\perp = d_4 - \mathbf{V}\psi_4,$$

$$\lambda_4 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\psi_4, \quad \lambda_5 = \varphi_1(\tfrac{2}{3}h\mathbf{H})\psi_4, \quad \lambda_6 = \varphi_1(h\mathbf{H})\psi_4,$$

$$k_4 = \mathbf{V}\lambda_4 + d_4^\perp, \quad k_5 = \mathbf{V}\lambda_5 + d_4^\perp, \quad k_6 = \mathbf{V}\lambda_6 + d_4^\perp, \tag{18d}$$

$$w_7 = \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6), \tag{18e}$$

$$u_7 = y_n + hw_7, \quad d_7 = f(u_7) - f(y_n) - h\mathbf{J}_n w_7, \tag{18f}$$

$$\psi_7 = \mathbf{V}^T d_7, \quad d_7^\perp = d_7 - \mathbf{V}\psi_7,$$

$$\lambda_7 = \varphi_1(\tfrac{1}{3}h\mathbf{H})\psi_7, \quad k_7 = \mathbf{V}\lambda_7 + d_7^\perp, \tag{18g}$$

$$y_1 = y_n + h\left(k_3 + k_4 - \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7\right). \tag{18h}$$

The single projection implementation of EROW4, called EROW4SP, is:

$$\psi_0 = \mathbf{V}^T f(y_n), \quad f_0^\perp = f(y_n) - \mathbf{V}\psi_0,$$

$$\lambda_1 = \varphi_1(\tfrac{1}{2}h\mathbf{H})f(y_n), \quad k_1 = \mathbf{V}\lambda_1 + f_0^\perp, \tag{19a}$$

$$w_2 = \frac{1}{2}k_1, \tag{19b}$$

$$u_2 = y_n + hw_2, \quad d_2 = f(u_2) - f(y_n) - h\mathbf{J}_n w_2, \tag{19c}$$

$$\psi_2 = V^T d_2, \quad d_2^\perp = d_2 - \mathbf{V}\psi_2,$$

$$\lambda_2 = \varphi_1(h\mathbf{H})\psi_0, \quad k_2 = \mathbf{V}\lambda_2 + f_0^\perp,$$

$$\lambda_3 = \varphi_1(h\mathbf{H})\psi_2, \quad k_3 = \mathbf{V}\lambda_3 + d_2^\perp, \tag{19d}$$

$$w_4 = k_2 + k_3, \tag{19e}$$

$$u_4 = y_n + hw_4, \quad d_4 = f(u_4) - f(y_n) - h\mathbf{J}_n w_4, \tag{19f}$$

$$\psi_4 = V^T d_4, \quad d_4^\perp = d_4 - \mathbf{V}\psi_4,$$

$$\lambda_4 = \varphi_3(h\mathbf{H})d_2, \quad k_4 = \mathbf{V}\lambda_4 + \frac{1}{3!}d_2^\perp, \quad \lambda_5 = \varphi_4(h\mathbf{H})d_2, \quad k_5 = \mathbf{V}\lambda_5 + \frac{1}{4!}d_2^\perp,$$

$$\lambda_6 = \varphi_3(h\mathbf{H})d_4, \quad k_6 = \mathbf{V}\lambda_6 + \frac{1}{3!}d_4^\perp, \quad \lambda_7 = \varphi_4(h\mathbf{H})d_4, \quad k_7 = \mathbf{V}\lambda_7 + \frac{1}{4!}d_4^\perp, \tag{19g}$$

$$y_{n+1} = y_n + h(k_2 + 16k_4 - 48k_5 - 2k_6 + 12k_7). \tag{19h}$$

5.4. Accuracy analysis of alternative implementations

Using the approach described by Algorithm 2 we construct B-series representations of the numerical solutions produced by EXP4K, EXP4SP, EROW4K, and EROW4SP. Table 2 shows the B-series coefficients for up to fourth order. Note that coefficients associated to various trees change not only for different methods but also for different formulations of the same method.

The critical coefficient is that belonging to τ_{13}, i.e., corresponding to the K order condition. Table 2 reveals that EXP4K is fourth order, while both EXP4SP, EROW4K, and EROW4SP are only third order. These analytical results are confirmed experimentally in the next section.

i	$F(\tau_i)$	EXP4K	EXP4SP	EROW4K	EROW4SP	Exact Solution
1	f^J	1	1	1	1	1
2	$f_K^J f^K$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$
3	$\mathbf{A}_{JK} f^K$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	
4	$f_{KL}^J f^K f^L$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
5	$f_K^J f_L^K f^L$	$\frac{1}{6}$	0	$\frac{1}{12}$	0	$\frac{1}{6}$
6	$f_K^J \mathbf{A}_{KL} f^L$	$\frac{1}{120}$	0	$\frac{1}{12}$	0	
7	$\mathbf{A}_{JK} f_L^K f^L$	$-\frac{1}{36}$	0	$\frac{1}{15}$	0	
8	$\mathbf{A}_{JK} \mathbf{A}_{KL} f^L$	$\frac{7}{360}$	$\frac{1}{6}$	$-\frac{1}{15}$	$\frac{1}{6}$	
9	$f_{KLM}^J f^K f^L f^M$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
10	$f_{KL}^J f_M^L f^M f^K$	$\frac{1}{6}$	0	$\frac{1}{12}$	0	$\frac{1}{8}$
11	$f_{KL}^J \mathbf{A}_{LM} f^M f^K$	$-\frac{1}{24}$	$\frac{1}{8}$	$\frac{1}{24}$	$\frac{1}{8}$	
12	$f_K^J f_{LM}^K f^M f^L$	$\frac{1}{12}$	0	$\frac{1}{24}$	0	$\frac{1}{12}$
13	$\mathbf{A}_{JK} f_{LM}^K f^M f^L$	0	$\frac{1}{12}$	$\frac{1}{24}$	$\frac{1}{12}$	0
14	$f_K^J f_L^K f_M^L f^M$	0	0	0	0	$\frac{1}{24}$
15	$f_K^J f_L^K \mathbf{A}_{LM} f^M$	$\frac{1}{20}$	0	$\frac{1}{48}$	0	
16	$f_K^J \mathbf{A}_{KL} f_M^L f^M$	$\frac{1}{18}$	0	$\frac{1}{24}$	0	
17	$f_K^J \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$	$-\frac{1537}{24300}$	0	$-\frac{1}{48}$	0	
18	$\mathbf{A}_{JK} f_L^K f_M^L f^M$	$\frac{1}{36}$	0	$\frac{1}{120}$	0	
19	$\mathbf{A}_{JK} f_L^K \mathbf{A}_{LM} f^M$	$-\frac{23}{720}$	0	$\frac{1}{80}$	0	
20	$\mathbf{A}_{JK} \mathbf{A}_{KL} f_M^L f^M$	$-\frac{1}{27}$	0	$-\frac{1}{60}$	0	
21	$\mathbf{A}_{JK} \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$	$\frac{3943}{97200}$	$\frac{1}{24}$	$-\frac{1}{240}$	$\frac{1}{24}$	

Table 2: B-series expansion of the numerical solution for different exponential methods and implementations.

	Standard	K-type	SP-type
EXPK	–	3.99	–
EXP4	3.98	3.97	2.97
EROW4	4.00	2.97	2.96

Table 3: Convergence rates for all methods and implementations applied to the Lorenz-96 model.

6. Numerical Results

We perform numerical tests using a nonlinear ODE model, the two-dimensional shallow water equations, and the two-dimensional Allen-Cahn problem. While we have constructed both K- and SP- type implementations of EXP4 and EROW4, we present performance comparisons of only the fourth order methods EXPK, EXP4, EXP4K, and EROW4, since the third order methods perform the same amount of work as fourth order methods but yield lower accuracy.

6.1. Lorenz-96 model

The chaotic Lorenz-96 model [12] has $N = 40$ states, periodic boundary conditions, and is described by the following equations:

$$
\begin{aligned}
\frac{dy_j}{dt} &= -y_{j-1}\left(y_{j-2} - y_{j+1}\right) - y_j + F , \quad j = 1, \ldots, N , \\
y_{-1} &= y_{N-1} , \quad y_0 = y_N , \quad y_{N+1} = y_1 .
\end{aligned}
\tag{20}
$$

The forcing term is $F = 8.0$, with $t \in [0, \ 0.3]$ (time units).
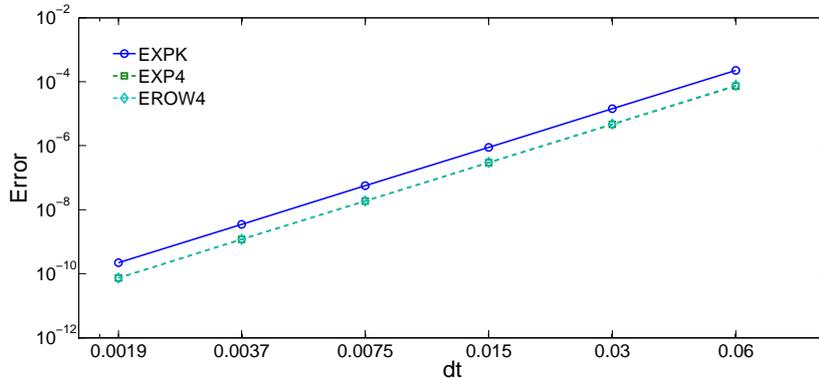
Figure 4: Precision diagrams for different exponential methods in standard implementation applied to the Lorenz-96 test problem. EXPK use a Krylov space of dimension $M = 5$.

Figure 4 shows the precision diagrams for EXPK with $M = 5$, and for the standard implementations of EXP4 and EROW4. All methods show the theoretical convergence order four. The performance of different implementations of EXP4 and EROW4 are shown in Figures 5(a) and 5(b), respectively. The results confirm the lower orders of alternative implementations predicted by the B-series analysis presented in table 2. The convergence rates for all methods applied to the Lorenz-96 model are summarized in Table 3 .

We examine the relative performance of the methods on the two-dimensional shallow water equations [10], a hyperbolic system of partial differential equations

$$\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) = 0, \tag{21a}$$

$$\frac{\partial}{\partial t}(uh) + \frac{\partial}{\partial x}\left(u^2h + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(uvh) = 0, \tag{21b}$$

$$\frac{\partial}{\partial t}(vh) + \frac{\partial}{\partial x}(uvh) + \frac{\partial}{\partial y}\left(v^2h + \frac{1}{2}gh^2\right) = 0, \tag{21c}$$

where $u(x,y,t)$, $v(x,y,t)$ are the flow velocity components and $h(x,y,t)$ is the fluid height. After spatial discretization using centered finite differences (21) is brought to the standard ODE form (1) with

$$y = [u \ v \ h]^T \in \mathbb{R}^N, \quad f_y(t,y) = \mathbf{J} \in \mathbb{R}^{N \times N}.$$

The standard exponential integrators compute the product $\varphi(h\mathbf{A}_n)b$ with an adaptive basis size to guarantee accuracy and the comparisons include the cost of the extra residual computations required to do so, while the K- type implementations use a constant basis size chosen empirically for stability. Automatic selection of Krylov basis size for stability is an open problem, and is the subject of future work. A special subroutine was implemented to compute exact Jacobian-vector products using a matrix-free approach.

Figures 6(a) and 6(b) show a performance comparison of the standard and K-type implementations of EXPK, EXP4, and EROW4. Two grid sizes of 32×32 and 128×128 points are considered. In both cases the K-type implementations are more efficient for lower error tolerances, while the adaptivity of the standard implementations allows them to 'catch up' in performance as the errors decrease.

6.3. The Allen-Cahn problem

For further performance comparison we consider the two-dimensional Allen-Cahn system, a parabolic partial differential equation

$$\frac{\partial}{\partial t}u = \alpha\nabla^2 u + \gamma\left(u - u^3\right), \quad (x,y) \in [0,1] \times [0,1], \quad t \in [0,0.2], \tag{22}$$

with $\alpha = 0.1$ and $\gamma = 1.0$. The problem has homogeneous Neumann boundary conditions and the initial solution $u(t=0) = 0.4 + 0.1(x+y) + 0.1\sin(10x)\sin(20y)$. Unlike the shallow water equations, the reaction-diffusion problem (22) is stiff.

The standard implementations of various methods make use of an adaptive Krylov basis size while the K-type implementations use empirically selected basis sizes. Figures 7(a) and 7(b) compares the performance of the standard and K-type implementations for different problem sizes. We once again see a better efficiency of the K-type methods for lower error values, but with a much earlier break-even point for efficiency. This is due primarily to the spectrum of the diffusion operator, and the difference in stability and accuracy requirements between the shallow water and Allen-Cahn equations. In the case of Allen-Cahn the stability requirements are more strict than the accuracy considerations and so the multiple smaller projections of the standard implementation become more efficient than the single larger projection in the K-type method, even though the latter uses fewer overall basis vectors.

7. Conclusions

This work extends the K-method approach proposed in [24] to exponential integrators and develops the new family of exponential-K schemes. A rigorous framework for order conditions analysis is developed that accounts for both temporal truncation errors and Krylov approximation errors. We construct an exponential-K method based on the general form given in [6], and reformulate existing exponential methods in order to take advantage of the reduced workload permitted by the new analysis.
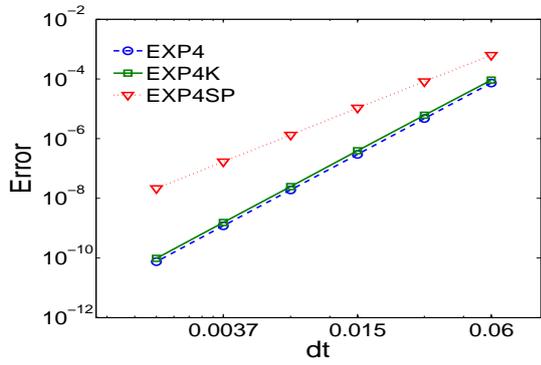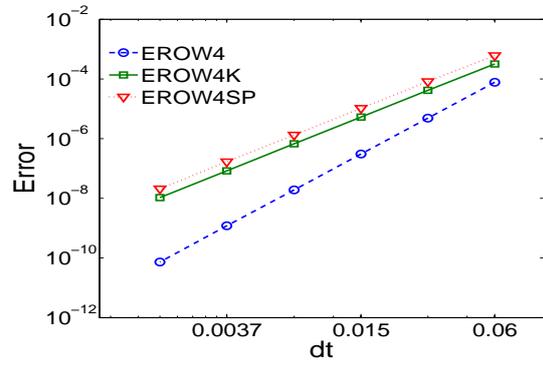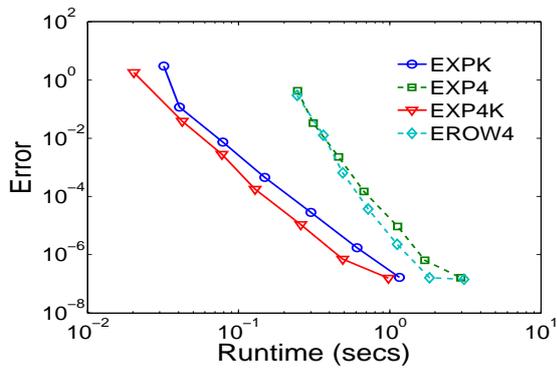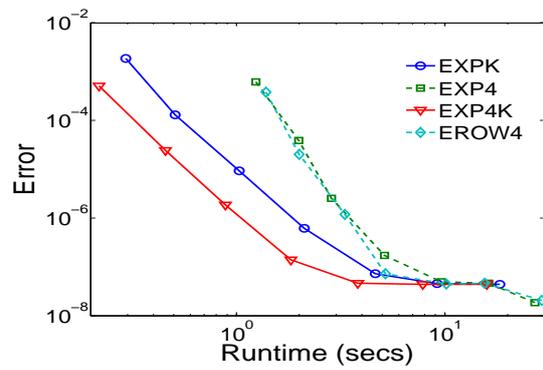
Figure 5: Work-precision diagrams for different implementations of traditional exponential integrators applied to the Lorenz-96 test problem (20).
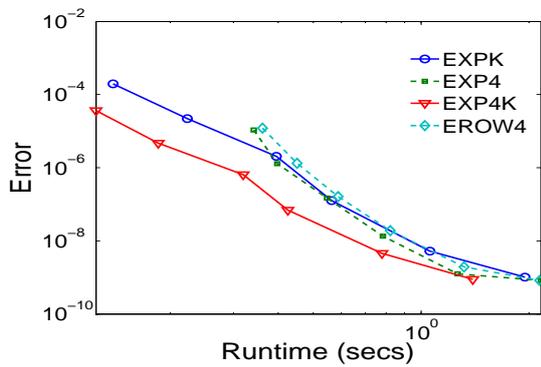

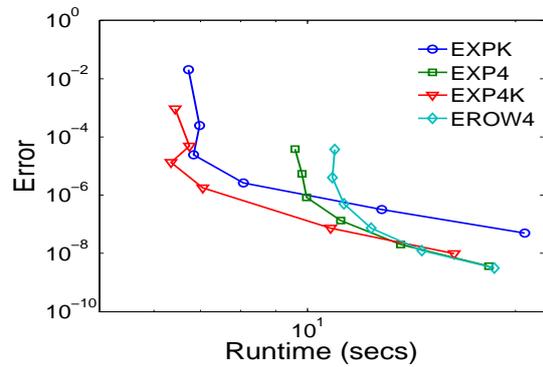Figure 6: Work-precision diagrams for exponential integrators applied to the shallow water test problem (21). Different problem sizes results from different spatial resolutions.

Figure 7: Work-precision diagrams for exponential integrators applied to the Allen-Cahn test problem (22). Different problem sizes result from different spatial resolutions.

Numerical experiments are carried out with three test problems, an ordinary differential equation and hyperbolic and parabolic partial differential equations. The results indicate that the new K-type exponential methods have the potential to be more efficient than their classical counterparts. While the new K- method EXPK derived here does not appear to be more efficient than the previously existing methods, primarily due to a less efficient general form, it validates the order conditions theory of exponential-K methods. We have shown that the traditional EXP4 method satisfies the additional order four K-condition when reformulated as a K-method, and that the resulting EXP4K scheme is more efficient than previous methods for the test problems presented here.

Future work will focus on developing a methodology to automatically select the Krylov subspace size in order to guarantee numerical stability, as on the construction of new exponential methods that can take full advantage of the inherent benefits present in the K-type formulation.

Acknowledgements

Appendix A. Order conditions for exponential-W methods.

Theorem 2 (Order conditions for exponential-W methods). *An exponential-W method with general form (2) has order p iff the following order conditions hold:*

$$\sum_j b_j \, \Phi_j(\tau) = P_\tau(\tau) \quad \forall \, \tau \in TW \quad with \quad |\tau| \leq p \, . \tag{A.1}$$

Here $|\tau|$ is the order, or number of vertices of the tree τ, and $\Phi_j(\tau)$ and $P_\tau(\tau)$ can be computed using Algorithm 2; they are shown in Figure A.4 for $p \leq 4$.

PROOF. The proof follows from our discussion in section 3, the near equivalence of order conditions for exponential-W and Rosenbrock-W methods [6], and from the order conditions of Rosenbrock-W methods [4, Theorem 7.7]. □

References

[1] J. C. Butcher. Trees, B-series and exponential integrators. IMA Journal of Numerical Analysis, 30(1):131–140, 2010.

[2] Philippe Chartier, Ernst Hairer, and Gilles Vilmart. Algebraic structures of B-series. Found. Comput. Math., 10(4):407–427, August 2010.

[3] E. Hairer, S.P. Nørsett, and G. Wanner. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer, 2000.

[4] E. Hairer and G. Wanner. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer, 2002.

[5] N. Higham. The scaling and squaring method for the matrix exponential revisited. SIAM Journal on Matrix Analysis and Applications, 26(4):1179–1193, 2005.

[6] Marlis Hochbruck, Christian Lubich, and Hubert Selhofer. Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput., 19(5):1552–1574, September 1998.

[7] Marlis Hochbruck, Alexander Ostermann, and Julia Schweitzer. Exponential Rosenbrock-type methods. SIAM J. Numer. Anal., 47(1):786–803, February 2009.

[8] C.T. Kelley, I.G. Kevrekidis, and L. Qiao. Newton-Krylov solvers for time-steppers. arXiv:math/0404374v1, 2004.

[9] D.A. Knoll and D.E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. Journal of Computational Physics, 193:357–397, 2004.

[10] Richard Liska and Burton Wendroff. Composite schemes for conservation laws, 1997.

[11] J. Loffeld and M. Tokman. Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs. Journal of Computational and Applied Mathematics, 241(0):45 – 67, 2013.

[12] Edward N.. Lorenz. Predictability – a problem partly solved. In Predictability of Weather and Climate. Cambridge University Press, 2006.

i	$F(\tau_i)$	$\Phi(\tau)$	$P_\tau(\gamma)$
1	f^J	1	1
2	$f^J_K f^K$	$\sum \alpha_{jk}$	$\frac{1}{2}$
3	$\mathbf{A}_{JK} f^K$	$\sum \gamma_{jk}$	$\frac{-\gamma}{2}$
4	$f^J_{KL} f^K f^L$	$\sum \alpha_{jk}\alpha_{jl}$	$\frac{1}{3}$
5	$f^J_K f^K_L f^L$	$\sum \alpha_{jk}\alpha_{kl}$	$\frac{1}{6}$
6	$f^J_K \mathbf{A}_{KL} f^L$	$\sum \alpha_{jk}\gamma_{kl}$	$\frac{-\gamma}{4}$
7	$\mathbf{A}_{JK} f^K_L f^L$	$\sum \gamma_{jk}\alpha_{kl}$	$\frac{-\gamma}{4}$
8	$\mathbf{A}_{JK} \mathbf{A}_{KL} f^L$	$\sum \gamma_{jk}\gamma_{kl}$	$\frac{\gamma^2}{3}$
9	$f^J_{KLM} f^K f^L f^M$	$\sum \alpha_{jk}\alpha_{jl}\alpha_{jm}$	$\frac{1}{4}$
10	$f^J_{KL} f^L_M f^M f^K$	$\sum \alpha_{jk}\alpha_{jl}\alpha_{jm}$	$\frac{1}{8}$
11	$f^J_{KL} \mathbf{A}_{LM} f^M f^K$	$\sum \alpha_{jk}\alpha_{jl}\gamma_{lm}$	$\frac{-\gamma}{6}$
12	$f^J_K f^K_{LM} f^M f^L$	$\sum \alpha_{jk}\alpha_{kl}\alpha_{km}$	$\frac{1}{12}$
13	$\mathbf{A}_{JK} f^K_{LM} f^M f^L$	$\sum \gamma_{jk}\alpha_{kl}\alpha_{km}$	$\frac{-\gamma}{6}$
14	$f^J_K f^K_L f^L_M f^M$	$\sum \alpha_{jk}\alpha_{kl}\alpha_{lm}$	$\frac{1}{24}$
15	$f^J_K f^K_L A_{LM} f^M$	$\sum \alpha_{jk}\alpha_{kl}\gamma_{lm}$	$\frac{-\gamma}{12}$
16	$f^J_K \mathbf{A}_{KL} f^L_M f^M$	$\sum \alpha_{jk}\gamma_{kl}\alpha_{lm}$	$\frac{-\gamma}{12}$
17	$f^J_K \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$	$\sum \alpha_{jk}\gamma_{kl}\gamma_{lm}$	$\frac{\gamma^2}{6}$
18	$\mathbf{A}_{JK} f^K_L f^L_M f^M$	$\sum \gamma_{jk}\alpha_{kl}\alpha_{lm}$	$\frac{-\gamma}{12}$
19	$\mathbf{A}_{JK} f^K_L \mathbf{A}_{LM} f^M$	$\sum \gamma_{jk}\alpha_{kl}\gamma_{lm}$	$\frac{\gamma^2}{8}$
20	$\mathbf{A}_{JK} \mathbf{A}_{KL} f^L_M f^M$	$\sum \gamma_{jk}\gamma_{kl}\alpha_{lm}$	$\frac{\gamma^2}{6}$
21	$\mathbf{A}_{JK} \mathbf{A}_{KL} \mathbf{A}_{LM} f^M$	$\sum \gamma_{jk}\gamma_{kl}\gamma_{lm}$	$\frac{-\gamma^3}{4}$

Table A.4: Order conditions for exponential-W methods with general form (2) and $p \leq 4$.

[13] Vu Thai Luan and Alexander Ostermann. Explicit exponential Runge Kutta methods of high order for parabolic problems. Journal of Computational and Applied Mathematics, 256(0):168 – 179, 2014.

[14] P. Novati. Some secant approximations for Rosenbrock W-methods. Applied Numerical Mathematics, 58(3):195–211, 2008.

[15] H. Podhaisky, R. Weiner, and B.A. Schmitt. Numerical experiments with Krylov integrators. Applied Numerical Mathematics, 28:413–425, 1997.

[16] G. Rainwater and M. Tokman. A new class of split exponential propagation iterative methods of rungekutta type (sEPIRK) for semilinear systems of ODEs. Journal of Computational Physics, 269(0):40 – 60, 2014.

[17] J. Rang and L. Angermann. New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1. BIT Numerical Mathematics, 45(4):761–787, 2005.

[18] R.Weiner, B.A. Schmitt, and H. Podhaisky. ROWMAP–a ROW-code with Krylov techniques for large stiff ODEs. Applied Numerical Mathematics, 25:303–319, 1997.

[19] Y. Saad. Iterative methods for sparse linear systems. PWS Pub. Co., Boston, 1996.

[20] B.A. Schmitt and R. Weiner. Matrix free W-methods using a multiple Arnoldi iteration. Applied Numerical Mathematics, 18:307–320, 1995.

[21] Roger B. Sidje. Expokit: a software package for computing matrix exponentials. ACM Trans. Math. Softw., 24(1):130–156, March 1998.

[22] M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. Journal of Computational Physics, 213(2):748 – 776, 2006.

[23] M. Tokman. A new class of exponential propagation iterative methods of Runge-Kutta type (EPIRK). J. Comput. Phys., 230(24):8762–8778, October 2011.

[24] Paul Tranquilli and Adrian Sandu. Rosenbrock-krylov methods for large systems of differential equations. SIAM J. Scientific Computing, 36(3), 2014.

[25] Henk A. van der Vorst. Iterative Methods for Large Linear Systems. Cambridge University Press, 2003.

[26] R. Weiner and B.A. Schmitt. Order results for Krylov-W methods. Computing, 61:69–89, 1998.