“Lightly Implicit Krylov-Exponential (LIKE) Methods”
Lightly Implicit Krylov-Exponential (LIKE) Methods

Paul Tranquilli*, Adrian Sandu*

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

Abstract
This paper develops a new class of exponential-type integrators where all the matrix exponentiations are performed in a single Krylov space of low dimension. The new family, called Lightly Implicit Krylov-Exponential (LIKE), is well suited for solving large scale systems of ODEs or semi-discrete PDEs. The time discretization and the Krylov space approximation are treated as a single computational process, and the Krylov space properties are an integral part of the new LIKE order condition theory developed herein. Consequently, LIKE methods require a small number of basis vectors determined solely by the temporal order of accuracy. The subspace size is independent of the ODE under consideration, and there is no need to monitor the errors in linear system solutions at each stage. Numerical results illustrate the favorable properties of new family of methods.

1. Introduction

Many methods exist to construct an approximate solution to the initial value problem

$$\frac{dy}{dt} = f(t,y), \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_0; \quad y(t), f(t,y) \in \mathbb{R}^N.$$ (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 an implicit method requires the solution to a linear, or non-linear, system of equations.

Much work has been done towards the acceleration of the solutions to these systems. The generalized minimal residual (GMRES) method is a Krylov based iterative solver, and is the standard approach for quickly constructing approximate solutions to linear systems arising throughout the integration of ODEs. Jacobian-free Newton-Krylov (JFNK) methods [7, 8] make use of a GMRES like solver within a newton iteration to solve the nonlinear equations arising from Runge-Kutta and multistep methods.

Rosenbrock [4] methods, 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

$$J = \left. \frac{\partial f}{\partial y} \right|_{y=y_0}$$

in the method itself. Due to the approximate nature of solutions coming from iterative methods, the explicit appearance of \(J\) causes order reduction unless the system solution is very accurate. For this reason Rosenbrock-W methods [4, 12, 13], an extension of Rosenbrock methods allowing for arbitrary approximations of the matrix \(J\) have been developed.

Krylov-ROW methods [13, 16, 17, 23] couple Rosenbrock-W methods with a Krylov based solver for the linear system arising therein. A multiple Arnoldi process is used to enrich the Krylov space at each stage,
and the order of the underlying Rosenbrock-W method is preserved with modest requirements on the Krylov space size, independent of the dimension of the ODE system under consideration.

Rosenbrock-K methods \cite{21} pursue the same goal of Krylov-ROW methods with one important difference, where Krylov-ROW methods give order results by constructing requirements for the Krylov space and making use of standard Rosenbrock-W methods, Rosenbrock-K methods guarantee an 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 methods of higher order, or fewer stages. More importantly Rosenbrock-K methods give a strict lower bound on the number of Krylov basis vectors required based only on the order of the method, and completely independent of the dimension of the ODE system being considered.

Exponential integrators \cite{5, 19, 20} replace the need to construct solutions to a linear system, or equivalently approximate the rational matrix function \((I_N - \tau A)^{-1}\), with the similar, hopefully cheaper, requirement to approximate the exponential matrix function \(\exp(\tau A)\). Where Krylov methods are standard in the pursuit of approximate solutions to linear systems, so too are Krylov based approximation methods a standard approach for the approximation of the matrix exponential.

Following class of exponential-W integrators proposed in \cite{5} we will examine the scheme:

\[
k_i = \varphi(h\gamma J) \left( hf(u_i) + hA \sum_{j=1}^{i-1} \gamma_{ij} k_j \right)
\]

\[
u_i = y_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j
\]

\[
y_1 = y_0 + \sum_{j=1}^{i} b_{i} k_{i}
\]

where \(A\) is either the matrix \(J\) or an approximation of it. Equation (2) encapsulates Runge-Kutta methods, when \(\varphi(z) = 1\), Rosenbrock methods, when \(\varphi(z) = \frac{1}{1-z}\), and exponential methods, when \(\varphi(z) = \frac{(e^z - 1)}{z}\). Note that similar to the Rosenbrock methods discussed before, equation (2) has an explicit appearance of the matrix \(J\), and so it is natural to explore conditions allowing for arbitrary approximations as in the case of Rosenbrock-W methods. A brief discussion of such methods and their order conditions is given in \cite{5}.

In this paper we will extend the ideas of the Rosenbrock-K methods presented in \cite{21} to integrators of the form (2), introducing a new family of exponential-K methods. These new methods require the construction of only a single Kyrlov basis for each timestep, as opposed to each stage in the case of standard exponential methods. Moreover, the dimension of this subspace need only be as large as the desired order of the method to ensure accuracy.

The paper is laid out as follows: in Section 2 we present the framework of exponential-K methods as well as the Krylov approximation of the Jacobian used, in Section 4 we construct, and extend, the order conditions of exponential-W methods to the new exponential-K methods and give details on how to construct these conditions using both trees and \(B\)-series, in Section 4 we outline the construction of a four stage, fourth order exponential-K method, and in Section 6 we present some numerical results.

2. Formulation of exponential-K methods

Exponential-K methods are an extension of Exponential-W methods with the same general form given in (2) which make use of a specific, Krylov based approximation of the Jacobian matrix. To begin we construct the \(M\)-dimensional Krylov space \(K_M\) where \(M \ll N\) and

\[
K_M = \text{span} \{ f_n, J f_n, J^2 f_n, \ldots, J^{M-1} f_n \}
\]

\[
K_{\tilde{M}} = \text{span} \{ v_1, v_2, \ldots, v_M \}
\]
using a modified Arnoldi iteration \cite{22}. The Arnoldi iteration returns two important matrices: the matrix

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

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

\[ H = V^T J V \in \mathbb{R}^{M \times M}. \] \hfill (4)

From these two matrices we construct \( A \), the Krylov-approximation matrix.

\[ A = V H V^T = V V^T J V V^T. \] \hfill (5)

**Lemma 1.**

\[ A^k = V H^k V^T \]

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

\[ A^1 = V H V^T = V H^1 V^T \]

next we assume that \( A^{k-1} = V H^{k-1} V^T \) and show that \( A^k = V H^k V^T \).

\[ A^k = A A^{k-1} = V H V^T (V H^{k-1} V^T) = V H (V^T V) H^{k-1} V^T \]

because \( V \) is an orthonormal matrix \( V^T V = I_N \), and so

\[ A^k = V H^k V^T \]

\[ \square \]

**Lemma 2.**

\[ \varphi_i(h\gamma A) = \frac{1}{i!}(I_N - V V^T) + V \varphi(h\gamma H) V^T \]

**Proof.** It is possible to expand \( \varphi_i(z) \) as a taylor series \cite{10}:

\[ \varphi_i(z) = \sum_{k=0}^{\infty} c_k \frac{z^k}{(i + k)!} \] \hfill (6)

We have then that

\[ \varphi_i(h\gamma A) = \frac{1}{i!} I_N + \sum_{k=1}^{\infty} \frac{(h\gamma)^k}{(i + k)!} A^k \]

after applying lemma \cite{1} we obtain

\[ \varphi_i(h\gamma A) = \frac{1}{i!} I_N + \sum_{k=1}^{\infty} \frac{(h\gamma)^k}{(i + k)!} VH^k V^T. \]

Similarly we can expand \( V \varphi_i(h\gamma H) V^T \) as

\[ V \varphi_i(h\gamma H) V^T = \frac{1}{i!}VV^T + \sum_{k=1}^{\infty} \frac{(h\gamma)^k}{(i + k)!} VH^k V^T, \]

taking the difference we see that

\[ \varphi_i(h\gamma A) - V \varphi_i(h\gamma H) V^T = \frac{1}{i!} (I_N - VV^T). \]
Finally we move $V \varphi_i(h \gamma H) V^T$ across the equality to obtain

$$\varphi_i(h \gamma A) = \frac{1}{i!} (I_N - V V^T) + V \varphi_i(h \gamma H) V^T$$ \hspace{1cm} (7)

\[\square\]

**Remark 1.** For the methods we present here, $i$ is either zero or one.

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

$$k_i = \sum_{\lambda_i \in K_M} V \lambda_i + \sum_{\mu_i \in K_M^\perp} V \mu_i, \quad f_i = \sum_{\psi_i \in K_M} V \psi_i + \sum_{\delta_i \in K_M^\perp} V \delta_i$$

where $V \lambda_i$ and $V \psi_i$ represent the pieces of $k_i$ and $f_i$ which reside in the Krylov subspace $K_M$, and similarly, $\mu_i$ and $\delta_i$ are the portions residing in the space orthogonal to $K_M$. Inserting equation (7) and the split forms of $k_i$ and $f_i$ into the general form (2) we obtain

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

leading to the reduced stage form

$$\lambda_i = \varphi(h \gamma H) \left( \psi_i + h H \sum_{j=1}^{i-1} \gamma_{ij} \lambda_j \right)$$ \hspace{1cm} (8)

where the full stage values can be recovered as

$$k_i = V \lambda_i + (f_i - V \psi_i).$$ \hspace{1cm} (9)

A single step of an Autonomous exponential-K method is given in algorithm 1.

**Algorithm 1** One step of an autonomous exponential-K integrator

1: Compute $H$ and $V$ using the $N$-dimensional Arnoldi process [22]

2: for $i = 1, \ldots, s$ do

\[\triangleright \text{For each stage, in succession}\]

$$f_i = f \left( y_n + \sum_{j=1}^{i-1} \alpha_{i,j} k_j \right)$$

$$\psi_i = V^T f_i$$

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

$$k_i = V \lambda_i + h (f_i - V \psi_i)$$

3: end for

4: $y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i$
Remark 2. Because the matrix $H$ is only $M \times M$, direct methods of computing the matrix function $\varphi(h\gamma H)$ are appropriate. In the case of Rosenbrock methods where $\varphi(z) = 1/(1 - z)$ an LU-decomposition can be used, or for exponential methods where $\varphi(z) = (e^z - 1)/z$ a Padé approximation is utilized. Further, the matrix function need only be evaluated a single time when the matrices $H$ and $V$ are constructed.

Remark 3. We have only given here an autonomous form of the exponential-$K$ method. A non-autonomous form is possible, though requires the construction of an extended system and Jacobian. A full treatment for the construction of such an extension is given in [21], along with an $N+1$ dimensional Arnoldi iteration to simplify the process.

3. Order Conditions

To construct accuracy conditions 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 the set of $T$-trees, that allow for two different colored nodes.

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

In the context of $TW$-(and $TK$)-trees a meagre, or solid, node represents an appearance of the exact Jacobian matrix $J$, while a fat, or empty, node represents the appearance of the approximate Jacobian matrix $A$. 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. Where the mapping $a : TW \cup \emptyset \rightarrow \mathbb{R}$ represents the series

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

Here the order, $|\tau|$, and the symmetry, $\sigma(\tau)$, of a tree are defined in the same way as in the literature for single colored trees [3, 20].

Because $K$-methods are an extension of $W$-methods it is necessary to first construct order conditions for the $W$-methods before using lemma 3 and lemma 4 to “recolor” the resultant trees and finally obtain the exponential-$K$ order conditions we seek. We follow a similar derivation procedure to that outlined in [14]. Throughout the derivation we will track the progress of several truncated B-series, in which we include all terms up to order four.

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

We begin the construction of order conditions for the $W$-method with a truncated $B$-series for $y_0$

$$B^{-1}(y(t_0), \tau) = a_0 = \{a_0(\emptyset) = 1, x_i = 0 \forall i = 1, \ldots, 21\},$$

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
### Figure 1: TW-trees to order 4

| $\tau$ | $j$ | $k$ | $l$ | $j$ | $k$ |
|--------|-----|-----|-----|-----|-----|
| $F(\tau)$ | $f^j$ | $f^k_{JK}$ | $A_{JK} f^L$ | $f^j_{KL} f^L_{LM} f^M$ | $f^j_{KL} f^L_{LM} f^M$ |
| $a(\tau)$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| $B^{-1}(h f(B(a,y)), \tau)$ | $1$ | $x_1$ | $0$ | $x_1^j$ | $x_2$ |
| $B^{-1}(h A B(a,y), \tau)$ | $0$ | $0$ | $x_3$ | $0$ | $0$ |
| $B^{-1}(\varphi(h A B(a,y), \tau)$ | $x_1$ | $x_2$ | $x_3 + c_1 x_1$ | | |

| $\tau$ | $\tau$ | $\tau$ | $\tau$ |
|--------|--------|--------|--------|
| $F(\tau)$ | $f^j_{KL} A_{LM} f^M$ | $f^j_{KL} A_{LM} f^M$ | $A_{JK} f^L_{LM} f^M$ | $f^j_{KL} f^L_{LM} f^M$ |
| $a(\tau)$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ |
| $h f(B(a,y))$ | $x_3$ | $0$ | $0$ | $x_1^j$ |
| $B^{-1}(h A B(a,y), \tau)$ | $0$ | $x_2$ | $x_3$ | $0$ |
| $B^{-1}(\varphi(h A B(a,y), \tau)$ | $x_6$ | $x_7 + c_1 x_2$ | $x_8 + c_1 x_3 + c_2 x_1$ | $x_9$ |

| $\tau$ | $\tau$ | $\tau$ | $\tau$ |
|--------|--------|--------|--------|
| $F(\tau)$ | $f^j_{KL} A_{LM} f^M$ | $f^j_{KL} A_{LM} f^M$ | $A_{JK} f^L_{LM} f^M$ | $f^j_{KL} f^L_{LM} f^M$ |
| $a(\tau)$ | $x_11$ | $x_12$ | $x_13$ | $x_14$ |
| $h f(B(a,y))$ | $x_1 x_3$ | $x_4$ | $0$ | $x_5$ |
| $B^{-1}(h A B(a,y), \tau)$ | $0$ | $0$ | $x_4$ | $0$ |
| $B^{-1}(\varphi(h A B(a,y), \tau)$ | $x_11$ | $x_12$ | $x_13 + c_1 x_4$ | $x_14$ |

| $\tau$ | $\tau$ | $\tau$ | $\tau$ |
|--------|--------|--------|--------|
| $F(\tau)$ | $f^j_{KL} A_{LM} f^M$ | $f^j_{KL} A_{LM} f^M$ | $A_{JK} f^L_{LM} f^M$ | $f^j_{KL} f^L_{LM} f^M$ |
| $a(\tau)$ | $x_16$ | $x_17$ | $x_18$ | $x_19$ |
| $h f(B(a,y))$ | $x_7$ | $x_8$ | $0$ | $0$ |
| $B^{-1}(h A B(a,y), \tau)$ | $0$ | $0$ | $x_5$ | $x_6$ |
| $B^{-1}(\varphi(h A B(a,y), \tau)$ | $x_16$ | $x_17$ | $x_18 + c_1 x_5$ | $x_19 + c_1 x_6$ |

| $\tau$ | $\tau$ | $\tau$ | $\tau$ |
|--------|--------|--------|--------|
| $F(\tau)$ | $A_{JK} A_{KL} A_{LM} f^M$ | $A_{JK} A_{KL} A_{LM} f^M$ | $A_{JK} A_{KL} A_{LM} f^M$ | $A_{JK} A_{KL} A_{LM} f^M$ |
| $a(\tau)$ | $x_21$ | $x_16$ | $x_17$ | $x_18 + c_1 x_5$ |
| $B^{-1}(h f(B(a,y)), \tau)$ | $0$ | $0$ | $x_5$ | $x_6$ |
| $B^{-1}(h A B(a,y), \tau)$ | $6$ | $x_8$ | $x_9 + c_1 x_6$ | $x_10 + c_1 x_7 + c_2 x_2$ |
| $B^{-1}(\varphi(h A B(a,y), \tau)$ | $x_21 + c_1 x_8 + c_2 x_3 + c_3 x_1$ | $x_19 + c_1 x_6$ | $x_20 + c_1 x_7 + c_2 x_2$ |
Algorithm 2 Construction of $B$-series of the numerical solution of an $s$ stage $W$-method

for $i = 1, \ldots, s$ do
  $u = a_0$
  for $j = 1, \ldots, i - 1$ do
    $u = u + \alpha_{ij} \cdot k_i$
  end for
  $q = B^{-1}(hf(B(u, y)), \tau)$
  for $j = 1, \ldots, i - 1$ do
    $q = q + \gamma_{ij} \cdot B^{-1}(hA B(k_i, y), \tau)$
  end for
  $k_i = B^{-1}(\varphi(h \gamma A) B(q, y), \tau)$
end for

$a_n = a_0$
for $i = 1, \ldots, s$ do
  $a_n = a_n + b_i \cdot k_i$
end for

solution $y(t_0 + h)$, note that the sum of two $B$-series is a $B$-series with coefficients equal to the sum of coefficients of the $B$-series being combined.

The order conditions of the $W$-method are, as usual, obtained by matching the $B$-series of the exact solution, $B^{-1}(y(t_0 + h), \tau)$, and the numerical solution, $B^{-1}(y_n, \tau)$ up to the 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 [21]. This extension allows us to “recolor” $TW$-trees which contain linear trees and sub-trees, substantially reducing the number of required conditions. This is done using lemmas 3 and 4 given here without proof.

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

$$A^k f_n = J^k f_n,$$

where $M = \text{dim}(K_M)$.

Lemma 4 (Property of elementary differentials using the approximation (5)). When the Krylov approximation matrix (5) is used, all linear $TW$-trees of order $k \leq M$ 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 in which the so called $W$-conditions arising from linear trees and sub-trees can be ignored since they are exact in the case where the Krylov approximation matrix is used.

Definition 1 (TK-trees).

$$TK = \{TW\text{-trees: no linear sub-tree has a fat root}\}$$

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

Figure 3 gives all $TK$-trees and order conditions of the exponential-$K$ method to order four. Note that there are only nine $TK$-trees as opposed to the original twenty $TW$-trees. We obtain only a single extra order condition over 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.
Figure 2: TK-trees and exponential-K conditions up to order four.

| $\tau$ | $F(\tau)$ | $\Phi(\tau)$ | $P_\tau(\gamma)$ |
|--------|---------|-------------|-----------------|
| $j$    | $f^j$   | 1           | 1               |
| $k$    | $f^K$   | $\sum \beta_{j,k}$ | $1/2(1 - \gamma)$ |
| $j$    | $f^K_j$ | $\sum \alpha_{j,k} \alpha_{j,l}$ | $1/3$ |
| $k$    | $f^K_j$ | $\sum \beta_{j,k} \beta_{k,l}$ | $1/3(1/2 - \gamma)(1 - \gamma)$ |
| $j$    | $f^K_j f^K_k f^K_L f^K_M$ | $\sum \alpha_{j,k} \alpha_{j,l} \alpha_{j,m}$ | $1/4$ |
| $k$    | $f^K_j f^K_k f^K_L f^K_M$ | $\sum \alpha_{j,k} \beta_{k,l} \alpha_{j,m}$ | $1/8 - \gamma/6$ |
| $j$    | $f^K_j f^K_k f^K_L f^K_M$ | $\sum \alpha_{j,k} \alpha_{k,m} \alpha_{k,l}$ | $1/12$ |
| $k$    | $A_{jK} f^K_j f^K_k f^K_L f^K_M$ | $\sum \gamma_{j,k} \alpha_{k,m} \alpha_{k,l}$ | $-\gamma/6$ |
| $j$    | $f^K_j f^K_k f^K_L f^K_M$ | $\sum \beta_{j,k} \beta_{k,l} \beta_{l,m}$ | $1/4(1/3 - \gamma)(1/2 - \gamma)(1 - \gamma)$ |
We give the system of nine non-linear equations arising from the order conditions of a four stage, fourth order exponential-K method. As before we consider the case where \( \gamma_{ij} = \gamma \) for all \( i \) and denote
\[
\beta_{ij} = \alpha_{ij} + \gamma_{ij}, \quad \beta'_{ij} = \sum_{j=1}^{i-1} \beta_{ij}.
\]

We give the system of nine non-linear equations arising from the order conditions of a four stage, fourth order exponential-K method in equation (12).

\[
\begin{align*}
(a) \quad & b_1 + b_2 + b_3 + b_4 & = & 1 \\
(b) \quad & b_2\beta'_2 + b_3\beta'_3 + b_4\beta'_4 & = & \frac{1}{2}(1 - \gamma) = p_{21}(\gamma) \\
(c) \quad & b_2\alpha^2_2 + b_3\alpha^2_3 + b_4\alpha^2_4 & = & \frac{1}{8} \\
(d) \quad & b_3(\beta_2 + \beta_3) + b_4(\alpha_2 + \alpha_3 + \beta_3) & = & \frac{1}{8}(1 - \gamma)(1 - \gamma) = p_{3,2}(\gamma) \\
(e) \quad & b_2\alpha^2_2 + b_3\alpha^2_3 + b_4\alpha^2_4 & = & \frac{1}{4} \\
(f) \quad & b_3\alpha_2(\alpha' + 2) + b_4(\alpha_2 + \alpha_3) & = & \frac{1}{8} - \frac{1}{6}\gamma = p_{4,2}(\gamma) \\
g_1 \quad & b_3\alpha_2(\alpha' + 2) + b_4(\alpha_2 + \alpha_3) & = & \frac{1}{12} \\
(g_2) \quad & b_3\gamma_2 + b_4(\gamma_2 + \gamma_3) & = & -\frac{1}{6}\gamma \\
(h) \quad & b_4\gamma_3 + b_5\gamma_2 & = & \frac{1}{4}\left(\frac{1}{3} - \gamma\right)(\frac{1}{2} - \gamma)(1 - \gamma) = p_{4,4}(\gamma)
\end{align*}
\]

If we now set
\[
p_{4,3}(\gamma) = \frac{1}{12} - \frac{1}{6}\gamma,
\]
we can follow exactly the solution procedure given in [21] for obtaining the ROK4a method, where we make use of the \( p_{i,j} \) given above, and as suggested in [5] to guarantee exact solutions for linear ODEs choose \( \gamma \) as the reciprocal of an integer. For the method given in table 3 we make the arbitrary choices that
\[
\gamma = \frac{1}{4}, \quad b_3 = 0, \quad \alpha_2 = 1, \quad \alpha_3 = \alpha_4 = \frac{1}{2}, \quad \beta_{ij} = \frac{1}{4}.
\]
\[
\alpha_2,1 = \frac{1}{4}, \quad \gamma_2,1 = \frac{7}{8}, \\
\alpha_3,1 = \frac{41}{80}, \quad \gamma_3,1 = \frac{1}{16}, \\
\alpha_3,2 = \frac{1}{80}, \quad \gamma_3,2 = 0, \\
\alpha_4,1 = \frac{1}{4}, \quad \gamma_4,1 = \frac{1}{32}, \\
\alpha_4,2 = \frac{1}{12}, \quad \gamma_4,2 = \frac{1}{24}, \\
\alpha_4,3 = \frac{1}{8}, \quad \gamma_4,3 = \frac{5}{12}.
\]

\[
\begin{array}{l}
\gamma = \frac{1}{4} \\
\alpha_2,1 = \frac{1}{4}, \quad \gamma_2,1 = \frac{7}{8}, \\
\alpha_3,1 = \frac{41}{80}, \quad \gamma_3,1 = \frac{1}{16}, \\
\alpha_3,2 = \frac{1}{80}, \quad \gamma_3,2 = 0, \\
\alpha_4,1 = \frac{1}{4}, \quad \gamma_4,1 = \frac{1}{32}, \\
\alpha_4,2 = \frac{1}{12}, \quad \gamma_4,2 = \frac{1}{24}, \\
\alpha_4,3 = \frac{1}{8}, \quad \gamma_4,3 = \frac{5}{12}.
\end{array}
\]

Table 1: Coefficients of expK4, a fourth order exponential-K method.

5. Alternative Krylov-Based Implementations of Existing Methods.

We present two new reformulations of previously derived methods exp4 [5] in equation (13) and erow4 [6] in equation (14). These reformulations make use of only a single Krylov subspace projection, exploiting the B-series analysis of section 3.

\[
k_1 = \varphi_1 \left( \frac{1}{3} hA \right) f(y_0), \quad k_2 = \varphi_1 \left( \frac{2}{3} hA \right) f(y_0), \quad k_3 = \varphi_1 (hA) f(y_0), \quad (13a)
\]

\[
w_4 = \frac{-7}{300} k_1 + \frac{97}{150} k_2 - \frac{37}{300} k_3, \quad (13b)
\]

\[
u_4 = y_0 + h w_4, \quad d_4 = f(u_4) - f(y_0) - hA w_4, \quad (13c)
\]

\[
k_4 = \varphi_1 \left( \frac{1}{3} hA \right) d_4, \quad k_5 = \varphi_1 \left( \frac{2}{3} hA \right) d_4, \quad k_6 = \varphi_1 (hA) d_4, \quad (13d)
\]

\[
w_7 = \frac{59}{300} k_1 - \frac{7}{75} k_2 + \frac{269}{300} k_3 + \frac{2}{3} (k_4 + k_5 + k_6), \quad (13e)
\]

\[
u_7 = y_0 + h w_7, \quad d_7 = f(u_7) - f(y_0) - hA w_7, \quad (13f)
\]

\[
k_7 = \varphi_1 \left( \frac{1}{3} hA \right) d_7, \quad (13g)
\]

\[
y_1 = y_0 + h \left( k_3 + k_4 - \frac{4}{3} k_5 + k_6 + \frac{1}{6} k_7 \right). \quad (13h)
\]
EROW4 has a similar form, modified from the source material:

\[ k_1 = \varphi_1 \left( \frac{1}{2} hA \right) f(y_0), \]  
\[ w_2 = \frac{1}{2} k_1, \]  
\[ u_2 = y_0 + h w_2, \quad d_2 = f(u_2) - f(y_0) - h A w_2, \]  
\[ k_2 = \varphi_1(hA)f(y_0), \quad k_3 = \varphi_1(hA)d_2, \]  
\[ w_4 = k_2 + k_3, \]  
\[ u_4 = y_0 + h w_4, \quad d_4 = f(u_4) - f(y_0) - h A w_4, \]  
\[ k_4 = \varphi_3(hA)d_2, \quad k_5 = \varphi_4(hA)d_2, \quad k_6 = \varphi_3(hA)d_4, \quad k_7 = \varphi_4(hA)d_4, \]  
\[ y_1 = y_0 + h (k_2 + 16k_4 - 48k_5 - 2k_6 + 12k_7). \]  

We implement these methods in three different forms: first, in the standard way outlined in the literature \[5, 19, 20\]; second, entirely in the reduced space such as given above and in \[21\]; and finally, using only a single Krylov projection to approximate the \( \varphi \) functions.

### 5.1. Standard Implementation

The primary feature of a standard implementation of an exponential method is the approximation of \( \varphi \) functions using Krylov subspaces. This is done for a term of the form \( \varphi(hA)b \) by projecting \( \varphi(hA) \) and \( b \) into the space \( \mathcal{K}_M(A, b) = \text{span} \{ b, A b, A^2 b, \ldots, A^{M-1} b \} \), using the projector matrix \( V V^T \) as in equation (15).

\[ \varphi(hA)b \approx V V^T f(hA) V V^T b \]  

Making use of equation (4), and noting that \( V^T b = \| b \|_2 e_1 \), we obtain the final Krylov subspace approximation

\[ \varphi(hA)b \approx \| b \|_2 V \varphi(hH)e_1 \]  

where \( e_1 \) is the first canonical basis vector. This approximation is computed as in \[18\], in which the exponential of an augmented matrix, \( \tilde{H} \), is constructed and \[16\] is read off from this result.

**Remark 5.** This implementation requires the construction of a new Krylov space for each vector operated on by a \( \varphi \) function, as well as the evaluation of a matrix exponential to compute each \( \varphi \) function. Both EXP4 and EROW4 require the construction of three Krylov subspaces, and the evaluation of seven matrix exponentials.

### 5.2. K-type Implementation

We implement EXP4 and EROW4 in the style of \[21\] and section 1. For each \( k_i \in \mathbb{R}^N \) we create a corresponding \( \lambda_i = V^T k_i \in \mathbb{R}^M \), similarly \( \sigma_i = V^T w_i \in \mathbb{R}^M \), and evaluate all linear algebra operations, including Jacobian-vector products in the reduced space. Further, we construct only a single Krylov subspace and compute the full matrix of three \( \varphi \) function evaluations in the case of EXP4, and four in the case of EROW4.
The $K$-type implementation of exp4, called exp4K, is given as:

$$
\psi_0 = V^T f(y_0), \quad f_0^+ = f - V\psi_0, \\
\lambda_1 = \varphi_1\left(\frac{1}{3}H\right)\psi_0, \quad \lambda_2 = \varphi_1\left(\frac{2}{3}H\right)\psi_0, \quad \lambda_3 = \varphi_1(hH)\psi_0, \\
k_1 = V\lambda_1 + f_0^+, \quad k_2 = V\lambda_2 + f_0^+, \quad k_3 = V\lambda_3 + f_0^+, \\
u_4 = \frac{-7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3, \quad \sigma_4 = \frac{-7}{300}\lambda_1 + \frac{97}{150}\lambda_2 - \frac{37}{300}\lambda_3, \\
u_4 = y_0 + hw_4, \quad \psi_4 = V^T f(u_4), \quad f_4^+ = f(u_4) - V\psi_4, \quad \delta_4 = \psi_4 - \psi_0 - hH\sigma_4, \\
\lambda_4 = \varphi_1\left(\frac{1}{3}H\right)\delta_4, \quad \lambda_4 = \varphi_1\left(\frac{1}{3}H\right)\delta_4, \quad \lambda_4 = \varphi_1\left(\frac{1}{3}H\right)\delta_4, \\
k_4 = V\lambda_4 + f_4^+ - f_0^+, \quad k_5 = V\lambda_5 + f_4^+ - f_0^+, \quad k_6 = V\lambda_6 + f_4^+ - f_0^+, \\
u_7 = \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6), \quad \sigma_7 = \frac{59}{300}\lambda_1 - \frac{7}{75}\lambda_2 + \frac{269}{300}\lambda_3 + \frac{2}{3}(\lambda_4 + \lambda_5 + \lambda_6), \\
u_7 = y_0 + hw_7, \quad \psi_7 = V^T f(u_7), \quad f_7^+ = f(u_7) - V\psi_7, \quad \delta_7 = \psi_7 - \psi_0 - hH\sigma_7, \\
\lambda_7 = \varphi_1\left(\frac{1}{3}H\right)\delta_7, \quad k_7 = V^T\lambda_7 + f_7^+ - f_0^+, \\
y_1 = y_0 + h(k_3 + k_4 + \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7). 
$$

Similarly we give erow4k as:

$$
\psi_0 = V^T f(y_0), \quad f_0^+ = f(y_0) - V\psi_0, \\
\lambda_1 = \varphi_1\left(\frac{1}{2}H\right)\psi_0, \quad k_1 = V\lambda_1 + f_0^+, \\
u_2 = \frac{1}{2}k_1, \quad \sigma_2 = \frac{1}{2}\lambda_1, \\
u_2 = y_0 + hw_2, \quad \psi_2 = V^T f(u_2), \quad f_2^+ = f(u_2) - V\psi_2, \quad \delta_2 = \psi_2 - \psi_0 - hH\sigma_2, \\
\lambda_2 = \varphi_1(hH)\psi_0, \quad k_2 = V\lambda_2 + f_0^+, \\
\lambda_3 = \varphi_1(hH)\delta_2, \quad k_3 = V\lambda_3 + f_2^+ - f_0^+, \\
u_4 = k_2 + k_3, \quad \sigma_4 = \lambda_2 + \lambda_3, \\
u_4 = y_0 + hw_4, \quad \psi_4 = V^T f(u_4), \quad f_4^+ = f(u_4) - V\psi_4, \quad \delta_4 = \psi_4 - \psi_0 - hH\sigma_4, \\
\lambda_4 = \varphi_3(hH)\delta_2, \quad k_4 = V\lambda_4 + \frac{1}{30}(f_2^+ - f_0^+), \\
\lambda_5 = \varphi_4(hH)\delta_2, \quad k_5 = V\lambda_5 + \frac{1}{30}(f_2^+ - f_0^+), \\
\lambda_6 = \varphi_3(hH)\delta_4, \quad k_6 = V\lambda_6 + \frac{1}{30}(f_4^+ - f_0^+), \\
\lambda_7 = \varphi_4(hH)\delta_4, \quad k_7 = V\lambda_7 + \frac{1}{30}(f_4^+ - f_0^+), \\
y_1 = y_0 + h(k_2 + 16k_4 - 48k_5 - 2k_6 + 12k_7). 
$$

5.3. Single Projection Implementation

The results of section 3 imply that contrary to the standard implementation in which several Krylov spaces are constructed, primarily due to the existence of a residual indicating how accurately the matrix function has been approximated, only a single Krylov subspace is required. Here we construct only the one subspace, and similarly to $K$-type implementation compute the full matrix result of the $\varphi$ functions. The
implementation differs from the $K$-type implementation in that the linear algebra, including Jacobian-vector products is computed in the fullspace.

**EXP4SP**

\[
\begin{align*}
\psi_0 &= V^T f(y_0), \quad f_0^+ = f(y_0) - V \psi_0, \\
\lambda_1 &= \varphi_1 \left( \frac{1}{3} H \right) \psi_0, \quad \lambda_2 = \varphi_1 \left( \frac{2}{3} H \right) \psi_0, \quad \lambda_3 = \varphi_1 (h H) \psi_0, \\
k_1 &= V \lambda_1 + f_0^+, \quad k_2 = V \lambda_2 + f_0^+, \quad k_3 = V \lambda_3 + f_0^+, \\
w_4 &= \frac{-7}{300} k_1 + \frac{97}{150} k_2 - \frac{37}{300} k_3, \\
u_4 &= y_0 + hw_4, \quad d_4 = f(u_4) - f(y_0) - hJw_4, \\
\omega_4 &= V^T d_4, \quad d_4^+ = d_4 - V \omega_4, \\
\lambda_4 &= \varphi_1 \left( \frac{1}{3} H \right) \omega_4, \quad \lambda_5 = \varphi_1 \left( \frac{2}{3} H \right) \omega_4, \quad \lambda_6 = \varphi_1 (h H) \omega_4, \\
k_4 &= V \lambda_4 + d_4^+, \quad k_5 = V \lambda_5 + d_4^+, \quad k_6 = V \lambda_6 + d_4^+, \\
w_7 &= \frac{59}{300} k_1 - \frac{7}{75} k_2 + \frac{269}{300} k_3 + \frac{2}{3} (k_4 + k_5 + k_6), \\
u_7 &= y_0 + hw_7, \quad d_7 = f(u_7) - f(y_0) - hJw_7, \\
\omega_7 &= V^T d_7, \quad d_7^+ = d_7 - V \omega_7, \\
\lambda_7 &= \varphi_1 \left( \frac{1}{3} H \right) \omega_7, \quad k_7 = V \lambda_7 + d_7^+, \\
y_1 &= y_0 + h \left( k_3 + k_4 - \frac{4}{3} k_5 + k_6 + \frac{1}{6} k_7 \right).
\end{align*}
\]

**EROW4SP**

\[
\begin{align*}
\psi_0 &= V^T f(y_0), \quad f_0^+ = f(y_0) - V \psi_0, \\
\lambda_1 &= \varphi_1 \left( \frac{1}{2} H \right) f(y_0), \quad k_1 = V \lambda_1 + f_0^+, \\
w_2 &= \frac{1}{2} k_1, \\
u_2 &= y_0 + hw_2, \quad d_2 = f(u_2) - f(y_0) - hJw_2, \\
\omega_2 &= V^T d_2, \quad d_2^+ = d_2 - V \omega_2, \\
\lambda_2 &= \varphi_1 (h H) \psi_0, \quad k_2 = V \lambda_2 + f_0^+, \\
\lambda_3 &= \varphi_1 (h H) \omega_2, \quad k_3 = V \lambda_3 + d_2^+, \\
w_4 &= k_2 + k_3, \\
u_4 &= y_0 + hw_4, \quad d_4 = f(u_4) - f(y_0) - hJw_4, \\
\omega_4 &= V^T d_4, \quad d_4^+ = d_4 - V \omega_4, \\
\lambda_4 &= \varphi_3 (h H) d_2, \quad k_4 = V \lambda_4 + \frac{1}{3!} d_2^+, \quad \lambda_5 = \varphi_4 (h H) d_2, \quad k_5 = V \lambda_5 + \frac{1}{4!} d_2^+, \\
\lambda_6 &= \varphi_3 (h H) d_4, \quad k_6 = V \lambda_6 + \frac{1}{3!} d_4^+, \quad \lambda_7 = \varphi_4 (h H) d_4, \quad k_7 = V \lambda_7 + \frac{1}{4!} d_4^+, \\
y_1 &= y_0 + h (k_2 + 16 k_4 - 48 k_5 - 2 k_6 + 12 k_7).
\end{align*}
\]

5.4. Analysis of alternate implementations.

Using a variant of algorithm 2, we construct the B-series representation of the numerical solution produced using the methods EXP4K, EXP4SP, EROW4K and EROW4SP. Table 2 shows the B-series coefficients up to fourth order for these methods. Notice that coefficients belonging to the various trees changes not only with the different choice of methods, but also with the different formulations of these methods.
The critical coefficient is that belonging to $\tau_{13}$, corresponding to the $K$ condition. From the table we can see that EXP4K is fourth order, while both EXP4SP, EROW4K, and EROW4SP are only third order. These analytical results are confirmed experimentally in table 3.

6. Numerical Results

6.1. Lorenz 96

The nonlinear test is carried out with the Lorenz-96 model \[11\]. This chaotic model has $N = 40$ states, periodic boundary conditions, and is described by the following equations:

$$ \begin{align*}
\frac{dy_j}{dt} & = -y_{j-1} (y_{j-2} - y_{j+1}) - y_j + F, \quad j = 1, \ldots, N, \\
y_{N+1} & = y_1, \quad y_0 = y_N, \quad y_{N+1} = y_1.
\end{align*} \tag{21} $$

The forcing term is $F = 8.0$, with $t \in [0, 0.3]$. 

| $i$ | $F(\tau_i)$ | EXP4K | EXP4SP | EROW4K | EROW4SP |
|-----|--------------|--------|--------|--------|--------|
| 1   | $f^J$        | 1      | 1      | 1      | 1      | 0      |
| 2   | $f^J f^K$    | $\frac{1}{2}$ | 0      | $\frac{1}{2}$ | 0      | 0      |
| 3   | $A_{JK} f^K$ | 0      | $\frac{1}{2}$ | 0      | $\frac{1}{2}$ | 0      |
| 4   | $f_{KL} f^K f^L$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 0      |
| 5   | $f_{KL} f^K f^L$ | $\frac{1}{6}$ | 0      | $\frac{1}{12}$ | 0      | 0      |
| 6   | $f_{KL} A_{KL} f^L$ | $\frac{1}{12}$ | 0      | $\frac{1}{12}$ | 0      | 0      |
| 7   | $A_{JK} f^K f^L$ | $\frac{1}{12}$ | 0      | $\frac{1}{12}$ | 0      | 0      |
| 8   | $A_{JK} A_{KL} f^L$ | $\frac{1}{12}$ | 0      | $\frac{1}{12}$ | 0      | 0      |
| 9   | $f_{KL} f^{LM} f^K f^M$ | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | 0      |
| 10  | $f_{KL} f^{LM} f^K f^M$ | $\frac{1}{6}$ | 0      | $\frac{1}{12}$ | 0      | 0      |
| 11  | $f_{KL} A_{LM} f^{FM} f^K$ | $-\frac{1}{24}$ | $\frac{1}{8}$ | $\frac{1}{24}$ | $\frac{1}{8}$ | 0      |
| 12  | $f_{KL} A_{LM} f^{FM} f^L$ | $\frac{1}{12}$ | 0      | $\frac{1}{24}$ | 0      | 0      |
| 13  | $A_{JK} f_{KL} f^{LM} f^M$ | 0      | $\frac{1}{12}$ | $\frac{1}{24}$ | 0      | 0      |
| 14  | $A_{JK} f_{KL} f^{LM} f^M$ | 0      | 0      | 0      | 0      | 0      |
| 15  | $f_{KL} f_{KL} A_{LM} f^M$ | $\frac{1}{26}$ | 0      | $\frac{1}{26}$ | 0      | 0      |
| 16  | $f_{KL} f_{KL} A_{LM} f^M$ | $\frac{1}{26}$ | 0      | $\frac{1}{26}$ | 0      | 0      |
| 17  | $f_{KL} f_{KL} A_{LM} f^M$ | $-\frac{1537}{24300}$ | 0      | $-\frac{1}{24}$ | 0      | 0      |
| 18  | $A_{JK} f_{KL} f_{KL} f^M$ | $\frac{1}{26}$ | 0      | $\frac{1}{26}$ | 0      | 0      |
| 19  | $A_{JK} f_{KL} f_{KL} f^M$ | $-\frac{23}{720}$ | 0      | $\frac{1}{60}$ | 0      | 0      |
| 20  | $A_{JK} f_{KL} f_{KL} f^M$ | $-\frac{1}{27}$ | 0      | $-\frac{1}{27}$ | 0      | 0      |
| 21  | $A_{JK} f_{KL} f_{KL} f^M$ | $\frac{3911}{97200}$ | $\frac{1}{24}$ | $-\frac{1}{24}$ | 0      | 0      |

Table 2: B-series expansion of the numerical solution using several differing methods and implementations.
Figure 3: Precision diagram for Lorenz 96, showing convergence order of methods with $M = 5$.

| Method | Standard | K-type | SP-type |
|--------|----------|--------|---------|
| EXPK   | 0        | 3.99   | 0       |
| EXP4   | 3.98     | 3.97   | 2.97    |
| EROW4  | 4.00     | 2.97   | 2.96    |

Table 3: Convergence rates for Lorenz-96 model

Figure 4: Precision diagram for Lorenz 96, showing convergence order of methods with $M = 5$. 
6.2. Shallow water equations

We examine the relative performance of the methods on the shallow water equations \[9\].

\[
\begin{align*}
\frac{\partial}{\partial t} h + \frac{\partial}{\partial x} (uh) + \frac{\partial}{\partial y} (vh) &= 0 \quad (22a) \\
\frac{\partial}{\partial t} (uh) + \frac{\partial}{\partial x} \left( u^2 h + \frac{1}{2} gh^2 \right) + \frac{\partial}{\partial y} (uh) &= 0 \quad (22b) \\
\frac{\partial}{\partial t} (vh) + \frac{\partial}{\partial x} (uvh) + \frac{\partial}{\partial y} \left( v^2 h + \frac{1}{2} gh^2 \right) &= 0, \quad (22c)
\end{align*}
\]

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 \[22\] is brought to the standard ODE form \[1\] with

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

The standard exponential integrators are implemented with adaptive selection of Krylov basis size, 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. Matrix-free Jacobian-vector products are computed exactly in a subroutine.

Figures \[6\] and \[7\] show a performance comparison of the standard, and \(K-\) type implementations of EXPK, EXP4, and EROW4 on grid sizes of \(32 \times 32\) and \(128 \times 128\) respectively. In both cases it is clear from the figures that the \(K-\) type implementations are more efficient for lower error tolerances, while the adaptivity of the standard implementations allows them to 'catch up' as the errors decrease.

6.3. Allen Cahn

We present a performance comparison on a reaction-diffusion problem with Neumann boundary conditions and initial value \(u(0) = 0.4 + 0.1(x + y) + 0.1 \sin(10x) \sin(20y)\) with \(t \in [0, 0.2]\).

\[
\frac{\partial}{\partial t} u = \nabla^2 u + u - u^3 \quad (23)
\]
Figure 6: Precision diagram for the shallow water equations, with $N = 3 \times 32 \times 32$.

Figure 7: Precision diagram for the shallow water equations, with $N = 3 \times 128 \times 128$. 
Similarly to the shallow water equation example, the standard methods make use of an adaptive Krylov basis size while the $K$-type implementation uses empirically selected basis sizes. Figures 8 and 9 show a performance comparison of the standard and $K$-type implementations of the integrators presented earlier. We once again see the improved 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 associated with 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 becomes more efficient than the single larger projection in the $K$-type method, even though the latter uses fewer overall basis vectors.

Figure 8: Precision diagram for the Allen Cahn equation with $N = 50 \times 50$.

Figure 9: Precision diagram for the Allen Cahn equation with $N = 150 \times 150$. 

18
7. Conclusions

We have presented an extension of K-methods to exponential integrators, constructed an exponential-K method based on the general form given in [5], outlined a new framework with which to analyze such methods, reformulated existing exponential methods to take advantage of the reduced workload permitted by the new analysis, and through the use of numerical experiments made the case that the new K-type exponential methods have the potential to be more efficient than their classical counterparts.

While the 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 did allow us to validate the theory of exponential-K methods. More importantly we have shown that exp4 satisfies the K-condition when reformulated as a K-method and that the method exp4k is more efficient than previous methods for the test problems presented here.

Future work will focus on a method of automatically choosing Krylov subspace size to guarantee stability, as well as the construction of new exponential methods that can optimally make use of the benefits present in the K-type formulation.

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. Norsett, 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] 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.
[6] Marlis Hochbruck, Alexander Ostermann, and Julia Schweitzer. Exponential Rosenbrock-type methods. SIAM J. Numer. Anal., 47(1):786–803, February 2009.
[7] C.T. Kelley, I.G. Kevrekidis, and L. Qiao. Newton-Krylov solvers for time-steppers. arXiv:math/0404374v1, 2004.
[8] 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.
[9] Richard Liska and Burton Wendroff. Composite schemes for conservation laws, 1997.
[10] 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.
[11] E. Lorenz. Predictability-a problem partly solved. In Predictability. Proceedings of the 1995 ECMWF Seminar, pages 1–18, 1996.
[12] P. Novati. Some secant approximations for Rosenbrock W-methods. Applied Numerical Mathematics, 58(3):195–211, 2008.
[13] H. Podhaisky, R. Weiner, and B.A. Schmitt. Numerical experiments with Krylov integrators. Applied Numerical Mathematics, 28:413–425, 1997.
[14] G. Rainwater and M. Tokman. A new class of split exponential propagation iterative methods of Runge-Kutta type (sEPIRK) for semilinear systems of ODEs. 2013.
[15] 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.
[16] 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.
[17] B.A. Schmitt and R. Weiner. Matrix free W-methods using a multiple Arnoldi iteration. Applied Numerical Mathematics, 18:307–320, 1995.
[18] Roger B. Sidje. Expokit: a software package for computing matrix exponentials. ACM Trans. Math. Softw., 24(1):130–156, March 1998.
[19] 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.
[20] M. Tokman. A new class of exponential propagation iterative methods of Runge-Kutta type (epirk). J. Comput. Phys., 230(24):8762–8778, October 2011.
[21] P. Tranquilli and A. Sandu. Rosenbrock-Krylov methods for large systems of differential equations. ArXiv e-prints, May 2013.
[22] Henk A. van der Vorst. Iterative Methods for Large Linear Systems. Cambridge University Press, 2003.
[23] R. Weiner and B.A. Schmitt. Order results for Krylov-W methods. Computing, 61:69–89, 1998.